UGC & Govt. Approved
Sonargaon University (SU)
সোনারগাঁও ইউনিভার্সিটি (এসইউ)
WE WILL RISE UP WE WILL SHINE

Computerized Sales and Inventory Management System

For the partial fulfilment for the award of the degree of
Bachelor of Engineering in
Computer Science And Engineering

Submitted By
Sharmin Akter
ID:CSE 1503006009
Section (6B)

Submitted To:
Md.Naderuzzaman
Assistant Professor
Department Of CSE
Sonargaon University,Dhaka

DECLARATION

I certify that

a. The work contained in the project is original and has been done by myself under the supervision of my supervisor.

b. The work has not been submitted to any other Institute for any degree or diploma.

c. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

d. Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

e. Whenever I have quoted written materials from other sources and due credit is given to the sources by citing them.

f. From the plagiarism test, it is found that the similarity index of whole thesis within 25% and single paper is less than 10 % as per the university guidelines.

Supervised By :

Md.Naderuzzaman
Assistant Professor
Department Of CSE

Submitted By :

Sharmin Akter
ID:CSE 1503006009
Section (6B)

Acceptance

This is certify that, I am writing to confirm my acceptance and this project has been carried out by me , **Sharmin Akter,** Student Of **CSE(6B),** ID :- **CSE1503006009** of **Sonargaon University** on May 17th 2019 and to tell you how delighted I am to be a part of fulfilment the requirement of CSE-400 project/thesis. I feel confident that she can make a significant contribution by this project .The report has been prepared by under my guidance and is a record of the bona fide work carried out successfully. The best of my knowledge and as per declaration ,no part of this project has been submitted anywhere for any degree ,Diploma or Certificate.

Now she is permitted to submit the project. I wish her all success in her future endeavours.

Project Supervisor



.....

Md.Naderuzzaman

Assistant Professor

Department Of CSE

SonargaonUniversity

Acknowledgement

Foremost, I would like to express my sincere gratitude to my honourable Teacher **Md. Naderuzzaman** for the continuous support of my study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and doing this project. I could not have imagined having a better advisor and mentor for my **BSc in CSE** study.

My sincere thanks also goes to all my fellows, and my seniors, for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last four years. Also I thank my friends of my class and honourable teacher for enlightening me the first glance of research.

Last but not the least, I would like to thank to all the family: parents for giving birth to all of us, this 6b family, at the first place and supporting me spiritually throughout my life.

Abstract

Inventory proportionality is the goal of demand-driven inventory management. The primary optimal outcome is to have the same number of days worth of inventory on handacross all products so that the time of run out of all products would be simultaneous.The secondary goal of inventory proportionality is inventory minimization. Byintegrating accurate demand forecasting with inventory management, replenishmentinventories can be scheduled to arrive just in time to replenish the product destined to runout first, while at the same time balancing out the inventory supply of all products tomake their inventories more proportional, and thereby closer to achieving the primary goal.

Accurate demand forecasting also allows the desired inventory proportions to bedynamic by determining expected sales out into the future; this allows for inventory to bein proportion to expected short-term sales or consumption rather than to past averages, amuch more accurate and optimal outcome.Integrating demand forecasting into inventory management in this way also allows for the prediction of the "can fit" point when inventory storage is limited on a per-product basis.

Existing System:

As we know the manual processing is quite tedious, time consuming, less accurate incomparison to computerized processing. Obviously the present system is not is exceptionconsultant encountering all the above problems.

1. Time consuming .
2. It is very tedious.
3. All information is not placed separately.
4. Lot of paper work.
5. Slow data processing.
6. Not user-friendly environment.
7. It is difficult to found records due file management system.
8.

Proposed System:

In new computerized system I tried to give these facilities.

1. Manually system changes into computerized system.
2. Friendly user interface.
3. Time saving.
4. Save paper work.
5. Connecting to database so we use different type of queries, data report

TABLE OF CONTENTS

CHAPTER Page

1 INTRODUCTION.....7
1.1 Motivation and background.....9
1.2 Aims and Objectives.....10
1.3 Scope and Limitation.....10
1.4 Definition of Terms.....11
2. Review of Related Literature.....13
2.1 Sales & inventory Monitoring Systems.....18
2.2 Checkout.....19
2.2 Management Software System Features.....19
3.Methodology.....21
3.1 Existing Theories and Ideas Associated.....22
3.2 Context diagram of the proposed system.....22
3.3 Methods and approach.....22
3.4 Tools and Models.....23
3.5 Budget Requirements.....23
3.6 Scheduling.....24
4. Analysis,Design and Development.....25
4.1 *System Analysis* 26
4.2 Algorithm..... 27
4.3 Flowchart 29
4.4. Front End Design..... 30
4.5 User End interface..... 31
4.6 Enter the Product..... 31
4.7 Enter Category..... 32
4.8 Enter Sub-category.....33
4.9 Enter product 33
5.0 Enter employee registration.....34
5.1 Login details.....34
5.2 Stockin for inventory.....34
5.3 Record panel.....34

5.3.3 Customer record……35

5.4 Additional tools……35

5.5 Billing……35

5.Source Codes…… 36

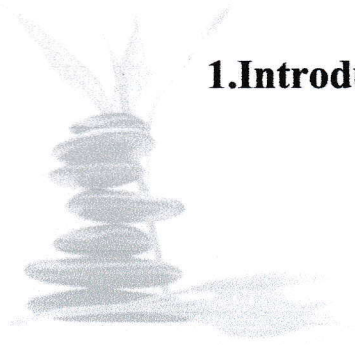
6. Summary, Conclusion and Recommendation…… 67

7.Reffference……70

8. Future Obectives……71

CHAPTER 01

1.Introduction



1.1 Motivation and Background

A sales and inventory management system is a software-based business solution used to simultaneously track sales activity and inventory. Manufacturers and trade resellers can both benefit from a thorough solution, where single transaction entry records necessary details on the customer, products purchased, price and date while also updating inventory levels.

Using computerized sales and inventory management systems allows for much greater accuracy in stocking and product management. They encourage ease of interaction between employees and shoppers as transactions are processed and items move from the business to the consumer. Computerized sales help provide better insight into which products are most popular. It also allows for enhanced marketing, stocking and oversight of critical sales objectives.

Computerized sales and inventory management systems save time for businesses by speeding up transactions while raising accuracy. This allows for confidence in accounting and accountability among employees as it is easy to verify how much money and what time transactions took place.

They also allow for consistent experiences in terms of customer service. People know there is always a uniform interaction at the register that requires tendering payment, taking a receipt and transition of ownership of products and services. This generates confidence in a business and ensures ongoing consumer relationships. Computerized systems are the most common method of inventory control and sales processing in retail markets.

In using this computerized sales and inventory management system you can establish the benefits you and your business handling may have. First is time savings. The amount of time that can be saved by a business is perhaps the biggest benefit of using a computerized inventory system. In cases where a shop maintains all data manually, its manager must reconcile each sales receipt with every piece of physical inventory. Depending on the size of the establishment and how many different products are sold, this can be a daunting and time consuming task. If that same store, however, used a computerized point of sale or POS System, the master inventory list would be updated electronically each time a sale is made. The only thing a manager would have to do each day is print out the report highlighting the inventory to be restocked. Second is accuracy. An additional benefit of using a computerized inventory management system is the accuracy it ensures. When an inventory list is maintained by hand, the margin of error widens with each update. If one mathematical calculation is wrong or one typo is made, disaster may occur. And lastly is the consistency. A small business operates most efficiently when its processes are executed in a consistent manner. By using a computerized inventory management system, a business owner

uniformity creates a professional appearance, which can go a long way to impress associates, such as potential investors.

Handling a whole business is a difficult job to do, especially if you are alone. The owner is the one who accepts the deliveries from the supplier, list the products delivered, arrange the product on the shelves and many more. So it is possible that the owner will get confused, especially if the customer and the supplier come in the same time. For example when there's a customer and a supplier came. The owner will get confused because he will attend to the needs of the customer while accepting the delivered products. Their ways of managing the sales and inventory is that they manually save records, accounts and their transaction which results to un-liquidated expenses. [1]Some expenses were not being recorded especially when the attendant or the owner has to get an emergency need for the business. Moreover, the expenses don't have any unsupported transparency records. Records don't co – inside the sales and the cash on hand. The given tally or the recorded amounts have their differences especially in the recorded amounts and the cash on hand.

1.2 Aims and Objectives

The proponents now endorse our proposal which can help to implement good and accurate system to the company. The proponents gave their best to satisfy the current needs of the business by making a computerized sales and inventory system that will help them in their business. Develop and implement computerized sales and inventory management system for Pharmacia Josefa Drugstore that will enhance the company's operation.

1.3 Scope and Limitation

The proponents proposed system for a companyaz can do file maintenance, product maintenance, transaction, product transaction, create an account and reports.

To design, develop and implement a module that will handle the security for the system.

User Accounts – this contains the function for adding, and editing name and password. Of course the authorized user can only be the one using the system.

Staff – can access only the inventory

To design, develop and implement a file maintenance module that will handle the files, records and important data of the business.

Products – It contains all the record of adding of products.

Employee – It contains all the record of adding and deleting of employees.

To design, develop and implement a sales module that will handle the sales transaction of their company.

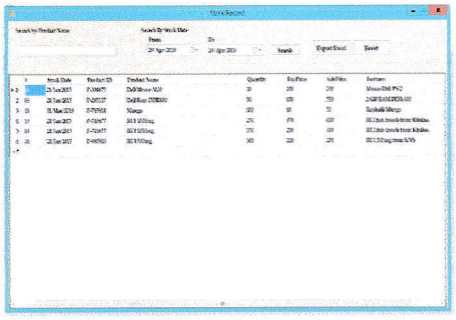
Purchased Order – a module that handles generating of purchase ordered for the supplier when they came.

Bad Order (customer) – list of items buys by the customer to the business, but needed to be return because of some damage product and must change with new ones but this must be within the store only.

Acquisition – list of items that are needed to be checks for possible damages, and to the company can return the damaged item to the specific supplier.

To design, develop and implement an inventory module that will handle the inventory of products. All items that are available in the company are monitored by the use of the following inventory methods:

Critical level – items that are low on stock will be monitored by the system in order to have sufficient stock.



To design, develop and implement sales module that will handle the inventory of products. This system provides sales transaction for the company. It holds information dateregarding their sales operations and provides an auto generated receipt for their customers.

To design, develop and implement a generated module that will handle the sales and inventory reports. This will provide the generation of reports which are necessary in proper evaluation of the company's sales and inventory records.

Maintenance

- Product Maintenance
- Employee Maintenance
- Supplier Maintenance
- Customer Maintenance

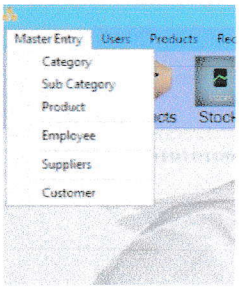


Fig : 03

Products

- Product Transaction
- Sales
- Stocks

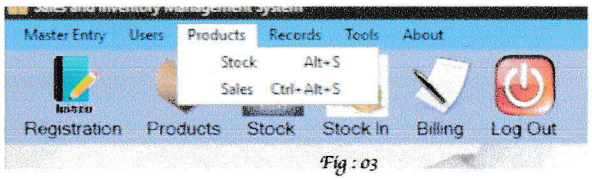


Fig : 03

Records

- Products
- Sales
- Stocks
- Product Supplier List
- Customer List
- Transaction Reports/Billing (Receipt)

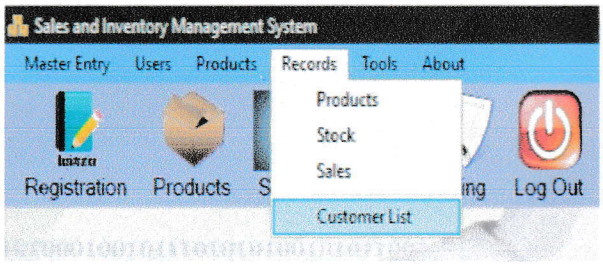


Fig : 03

The proponents proposed system for a company cannot transact with the use of credit card and cannot read or support barcodes and scanners.

1.4 Definition of Terms

Inventory – a complete list of the things that are in a place

Module – a computer program that does a particular job

Proponent – a person who supports something

Maintenance – the upkeep of property or equipment

Stocks – the supply of goods available for sale in a store

Products – something that is made or grown to be sold or used

Billing – the act of process of doing business with another person or company

POS- the time and place where retail transaction is completed and a point at which a customer makes a payment to the merchant in exchange for goods or after provision of a service

Employee – a person who works for another person or for a company for salary

Insufficient – not having or providing enough of what is needed

Inquiry – a request for information

Delivery – the act of taking something to a person or place

Security – the state of being protected or safe from harm

Staff – a group of people who work for a business

Purchase – to get something by paying money for it

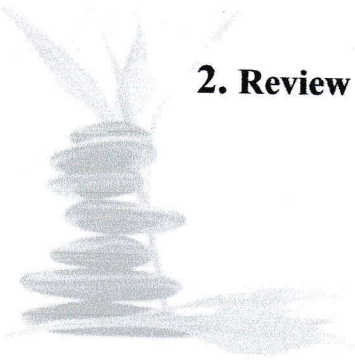
Order – a product/s that someone has requested from a business

Supplier – to provide someone or something that is needed or wanted

Accuracy– the ability to work or perform without making mistakes

Consistency – of the same quality or good each time

CHAPTER 02



2. Review of Related Literature

2. Review of Related Literature

Computer-based system is a complex system wherein information technology plays a major role. It makes the work easier, faster and more accurate. Due to that fact, the automated scheme has become essential to small and big companies for they are expected to give the best services possible. Nevertheless, some businesses still prefer sticking with the system that is not integrated with technology. Probable causes are computer illiterate staff and lack of funds. Companies, especially the big ones are recommended to switch from manual to automated systems because this will improve the efficiency and productivity of the business which will uplift the industry's reputation.

One of the most sought after automated systems of different companies is a sales and inventory system which comes hand in hand. A sales and inventory management system is very important in every organization because a good sales and inventory management can create excellent productivity. Primarily, inventory work consists of input, output and restock. Input is a process of buying new products into the inventory and replacing the old products with the new ones. Meanwhile, output is a procedure of taking out the products from the inventory for sales or usage and refill is a process of increasing the number of existing products in the inventory in order to fulfill the insufficient products or escalating demands. Most of the retailing market is using traditional way in the inventory management system where a person is assigned to check and record the stock by hand using pen and paper. It is where operations with regards to all the stock will be archived. It is without a doubt that one of the major roles played by today in almost every area in the society particularly in business and marketing is the computerized system. This system enables us to make very detailed work and follows accurate directives without error. The basic advantages of the system are to make the process fast and well-organized which means that they can process much more quickly than humans. Data retrieval has to be searched in lots of registers slowly and it wastes a lot of time. It makes the data not reliable enough as it is hand written and there is a high probability of errors to occur. Data recording consumes a lot of space since it is stored in cabinets and folders. It is also prone to data loss where it could be missing because of improper file management.

Inventory is a company's merchandise, goods and materials that are contained in the store factory at any given time. The employees need to know how many units of their products are available for reservation and sales and the items that are sold. All of these rely on the inventory system to present solutions. The sales inventory system would provide service to the user, input information to the database, summarize the inquiry of bills, and manage the product releases and storage. This would generate a faster improvisation of work given less time and effort. Inventory

contained within separate subsystems, but these functions must be performed in sequence in order to have a well-run inventory control system.

In today's business environment, even small and mid-sized businesses have come to rely on computerized inventory management systems. Certainly, there are plenty of small retail outlets, manufacturers, and other businesses that continue to rely on manual means of inventory tracking. Indeed, for some small businesses, like convenience stores, shoe stores, or nurseries, purchase of an electronic inventory tracking system might constitute a wasteful use of financial resources. But for other firms operating in industries that feature high volume turnover of raw materials and/or finished products, computerized tracking systems have emerged as a key component of business strategies aimed at increasing productivity and maintaining competitiveness. Moreover, the recent development of powerful computer programs capable of addressing a wide variety of record keeping needs—including inventory management—in one integrated system have also contributed to the growing popularity of electronic inventory control options

Given such developments, it is little wonder that business experts commonly cite inventory management as a vital element that can spell the difference between success and failure in today's keenly competitive business world.

Inventory is one of the most important in monitoring a stock that take place in business activity. The inventory system does the entire task in computing the value with inventory (cost and quality) and handling data or information. Inventory System maintains an orderly flow of supplies, raw materials, or finished goods through an office shop/factory because of items in any inventory. Represents cost, they need to be controlled. The purpose of inventory system for management are to keep inventory levels and cost at desire minimums while maintaining to proper safeguards over materials to places and people who need them.

Inventory means goods and materials, or those goods and materials themselves, held available in stock by a business. This word is also used for a list of the contents of a household and for a list for x purposes of the possessions of someone who has died. In accounting, inventory is considered an asset.

Automation can dramatically affect all phases of inventory management, including counting and monitoring of inventory items; recording and retrieval of item storage locations; recording changes to inventory; and anticipating inventory needs, including inventory handling requirements. Effective Inventory Analysis. Get Data Related to Effective Inventory Analysis. For many distributors, inventory is the largest and perhaps the most important asset. Inventory ties up more money than buildings or equipment, and is often less "liquid." It is crucial, then, that distributors develop and use a comprehensive set of tools that allows close monitoring of the performance of inventory investments.

record keeping chores would soon become overwhelming were it not for the computer and its unique information storage and retrieval capabilities.

According to the study of Anna Marie Beltran of Proposed Sales Monitoring System issued 1995, the accelerated work structure of the company proper monitoring is essential in order to keep track of the company's sales activities. Her system aims to come up with an efficient, and accurate mechanized system of preparing invoices that will keep track of the daily transaction and generate reports. Important information will be provided by this study in order to prove that the computerization of the company's operation and achieve a more reliable and efficient means of monitoring day to day activities. Her system focuses on the analysis and design of the sales monitoring system which will monitor the stacks inventory of the sales departments. This system of hers is somewhat related to our study but our system focuses more on the inventory monitoring system of a trading company.

According to Kj Henderson, they have a three major advantages by using of inventory system these are (Time savings, Accuracy, and Consistency) which helps a lot for a business man to increase the sales of their business.

- **Time Savings**

  As the old saying goes, "time is money". The amount of time that can be saved by a business is, perhaps, the biggest benefit of using a computerized inventory system. The only thing a manager would have to do each day is print out the report highlighting the inventory to be restocked.

- **Accuracy**

  An additional benefit of using a computerized inventory system is the accuracy it ensures. Eighteenth century English poet Alexander Pope is often quoted as having said, "To error is human." When an inventory list is maintained by hand, the margin of error widens with each update. If one mathematical calculation is wrong or one typo is made, disaster may occur. For instance, if a clerk accidentally adds a zero to the end of a purchase order, a business could potentially end up paying for 10,000 units of merchandise as opposed to the 1,000 that is actually needed.

- **Consistency**

  A small business operates most efficiently when its processes are executed in a consistent manner. By using a computerized inventory system, a business owner can ensures that all orders, reports and other documents relating to inventory are uniform in their presentation, regardless of who has created them.

Customer Needs, balancing the goals of avoiding stock outs while minimizing inventory costs is at the heart of just-in-time inventory. One of the main benefits of automated and efficient inventory replenishmentsystems is that you can quickly respond to reduced inventory levels. Companies are now equipped to pull back on stock in a given product category and ramp up inventory in another as customer needs and interests change.

Inventory , Costs Minimization of inventory management costs is a primary driver and benefit of just-in-time practices. You also have less likelihood of throwing out product that gets old or expires, meaning reduced waste.

Coordination, A disadvantage of managing a just-in-time inventory system is that it requires significant coordination between retailers and suppliers in the distribution channel. Retailers often put major trust in suppliers by syncing their computer systems with suppliers so they can more directly monitor inventory levels at stores or in distribution centres to initiate rapid response to low stock levels. This usually means build-up of technology infrastructure, which is costly. This coordinated effort is more involving on the whole than less time intensive inventory management systems.

Risks, just-in-time inventory is not without risks. By nature of what it is, companies using JIT intend to walk a fine line between having too much and too little inventory. If company buyers fail to adjust quickly to increased demand or if suppliers have distribution problems, the business risks upsetting customers with stock outs. If buyers over compensate and buy extra inventory to avoid stock outs, the company could experience higher inventory costs and the potential for waste.

Here are the following examples in the field of sales and inventory system.

2.1 Sales & Inventory Monitoring Systems

Features:

- The ability to record a sale against a customer and/or builder specifying line items, quantities, discount and GST applicable.
- The ability to receipt the sale as either an account or cash sale.
- The sales module must cater for quotes (provisional sales), credits and orders (pending assembly/delivery).
- The ability to merge detailed lines in an invoice to limit the granularity visible to the customer but still retain the detail for inventory and reporting/enquiry purposes.
- The ability to automatically generate a purchase order or purchase order reminder when

2.2 Checkout

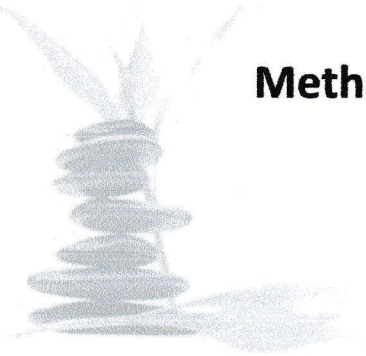
Features

- Allow user to enter item code either manually or automatically via barcode scanner. Automatic call up the standard or sales price, compute the price at multiple quantities and provide running total.
- Pricing – System includes add-on amounts, percentage of cost, margin percentage, and custom formulas.
- Inventory management
- Promotions such as discount, percentage off, time and date triggered

2.3 Management Software System Features

- Sales- It's where you take money from the customer in exchange for goods. The main function of your PMS is to automate the sales process and keep a detailed history of all transactions.
- Inventory Control- The second most important function of PMS software is to keep track of your inventory. Inventory control means knowing how much inventory you have and how much you need to buy at any given time.
- Masters- The master is where you will manage your entire repository about merchandising, distributor, hospital, doctor and employee information centrally to handle pharmacy operations more efficiently.
- CRM- CRM stands for Customer-Relations-Management, it's how you keep track of customers. Our PMS will help you store a list of customers, Track their purchase history for repeat sales and to conduct product promotions.

CHAPTER 03



Methodology

3.1 Existing Theories and Ideas Associated

This chapter presents theories associated in obtaining necessary data, design and development of the system. It involves discussion on the proposed conceptual framework diagram, the methods and approaches used the tools and models, budget requirements and the schedules involving the development of the system.

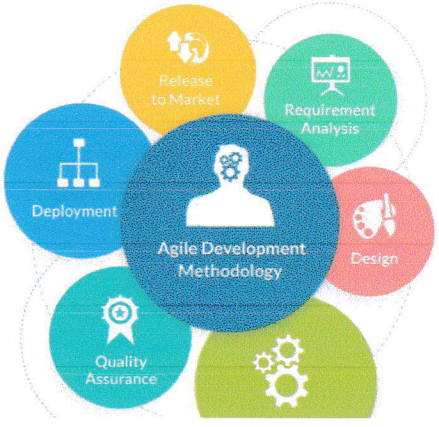
3.2 Context Diagram of the Proposed System



Figure 1. Context Diagram of the Proposed System

Figure 1 shows the context diagram depicting the system. This type of diagram usually pictures the system at the center, with no details of its interior structure, surrounded by all its interacting systems, environment and activities. The Diagram explains the main Input, Process and Output of the System where the Admin from the Input area in the left side decide to process his/her desired outcome. After it process in the System which illustrates at the center of this diagram, it barely results with an output of transaction consist of adding items. When the transaction process finished, the recorded files will be seen in the inventory that process again in the system that later develops a report back to the Admin.

3.2 Methods and Approaches



In the conduct of the system analysis and design, the proponents will use the agile software development method.

Agile software development is a group of software development methods in which solutions evolve through

to change. Further, the use of this method brings

clarity in the software development. Working software is delivered frequently (weeks rather than months). Regular adaptation to changing circumstances even late changes in requirements
are welcomed in this method we used.

3.4 Tools and Models

The proponent of this study uses the following tools and methods in the conduct of system analysis, as follows;

"The **Environmental Model** was used to describe or show the limitations and the interaction between our proposed system and the environment which it react. It focuses on the events that occur outside the environment and those that need a response from the system. In doing so thus eliminates irrelevant stimuli that occur outside the system. It is worthy to note the environment model is the most important part of making complete model, for it give a bird's eye view of the major steps and processes, and enable the user to visualize relationships, the important element and movement that exist in the system and to show the system react with the environment".

On the aspect of system development, the following tools were used;

Algorithm. An algorithm shows the proper sequence of steps that should be followed in order to obtain the required output.

Flow Chart. It is the graphical representation of the process that involved in the system. It has predefined graphics that describes the process the data involved to achieved desired output.

Visual Studio ,Visual studio 2017 was the final edition of the software. This version improved the productivity and ability for web applications, which led to the development of Visual Basic. NET. Visual Studio is capable to do develop desktop based software on Windows Vista, Windows 7,8,10 or Windows Server 2008 and Mac OS

Microsoft SQL Database Management Studio 2013, also known as MSSQL is a database management system from Microsoft that combines the relational Microsoft Jet Database Engine with a graphical user interface and software-development tools, It is a member of the Microsoft suite of application, include in the Professional and Higher editions or sold separately.

3.5Budget Requirements

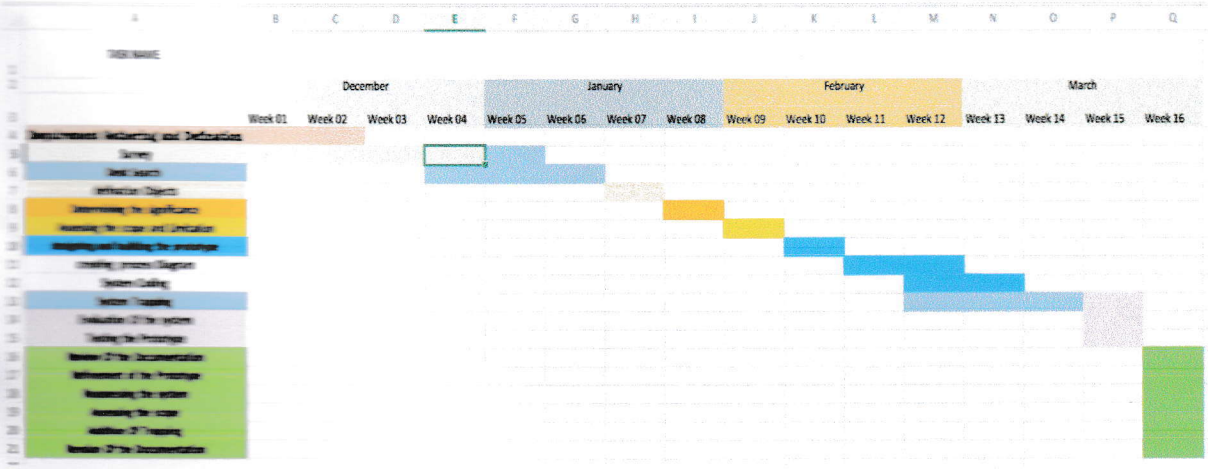
The following is the hardware and software specification necessary in the implementation of the system and necessary in the formulation of the essential budgetary requirements, as follows;

System Requirements	Price/Amount	Total Price
Receipt Printer	4,500 TK	4,500 TK
Desktop Unit	22,000 TK	12,000 TK
Bond paper (1 Rim)	150 TK	150 TK
Receipt paper roll	100TK	100 TK
Programmer	10,000 TK	10,000 TK
TOTAL:	-	**26,660**

3.6 Scheduling

The calendar of activities illustrates the software project schedule through a Gantt chart. This Chart were the days we plan to make and develop our research. As being planned, we decided to start this project on 02th December 2018 . In gathering data and our survey, we will make it within two weeks with schedule from our subject class hours. From 4up to 6we agreed to make 1 task per week. In designing and building of the prototype, we discussed that we'll be creating it for two weeks and a week after other task follow for the same number of days from WBS(Work BreakdownStructure) 2. Developing the System Code where a little bit difficult for a reason that we will work on it for 3 weeks to 4 weeks . The rest of the WBS, we decided to do it on the same remaining days. Figure shows the effort estimation of the present study.

Figure 2. Gantt Chart



CHAPTER 04

Analysis,Design and Development



This chapter presents the analysis, design and development of the desired system conducted by this study.

4.1. *System Analysis*

Event's List

1.0 Process Transaction

1.1 Add Transaction Information

1.1.1 by Product Name

1.1.2 By Category

1.1.3 By Sub Category

1.1.4 By Amount Received

1.1.5 By Customer Name

1.1.6 By Date

1.2 Delete Transaction Information

2.0 Process Payment

3.0 Process Inventory

Experimental Method

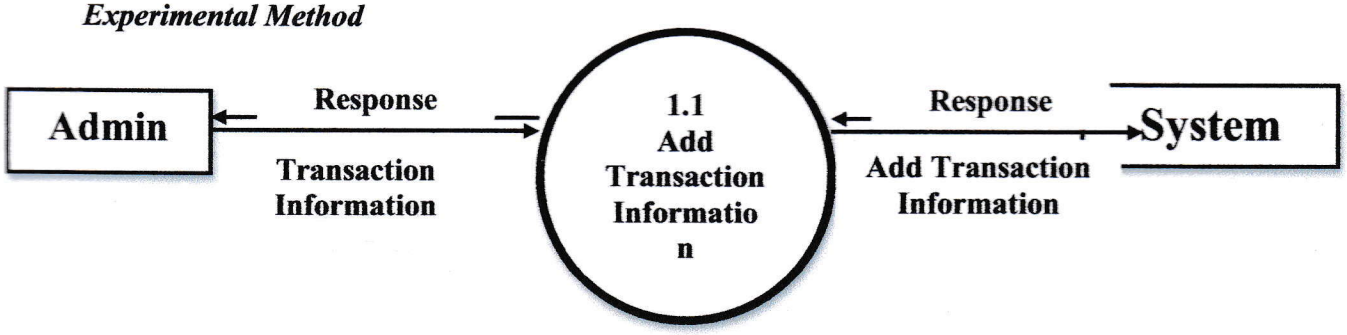


Figure .Add Transaction Informatio

In this diagram, nit discussed about the Admin, Transaction, and System. The Admin transact information and it processes on the System. Moreover, it is like a cycle where after it reach at the system it responses back to the Admin.



Figure 4. Process Payment

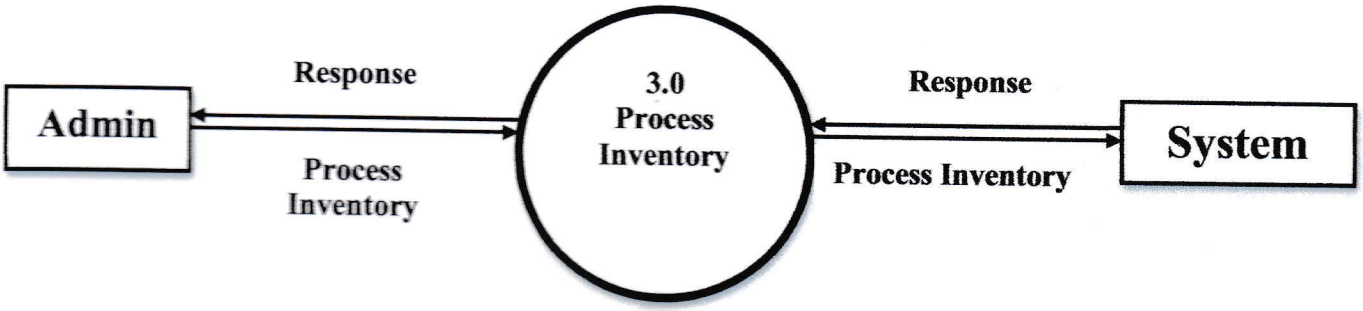


Figure 5. Process Inventory

In this diagram, it discussed about the Admin, Inventory, and System. The Admin process inventory and it proceeds to the System afterwards. When it reaches at the System it responses back to the Admin. Where the Admin can now view the Inventory Sales.

4.2 Algorithm

Step 1: Start of the Program

Step 2: Initialize new transaction and inventory sales

Step 3: Decide to update in new transaction and open this panel to insert product(if not exist)

Step 4: If yes: insert product name, productprice,subcategory,supplier with description of amount received . and select the customer ID (If exists) to give discount.

Step 5: If No: Click close

Step 6: Calculate the billing where product price * quantity and – amount received

Step 7: Click save and add this transaction to the cart

Step 9: Decide to save the transaction

Step 10: If Yes: Proceed to print

Step 11: If No: Close

Step 12: Decide to delete transaction

Step 13: If Yes: Print "Select all and count subtotal with vat and discount !!!"

Step 14: If No: Click to save data

Step 15: Stop the program.

Example :- supporse a customer affis resigtered is this sales and inventory system which is integreated to any store. For instance a customer wants take 1 bunch of banana then sales staff then first of all sales agent would have to check in the stock and is not then have to check with the suppliers.It its sounds that the suppliers provided the products but stock have not got any data yet.Sales Manager or Administrator have to put the data inside the stock.To sell or stock out,Billing panel is the panel related to inventory.where all the products can be found according to their name or id.Afterthat,sales staff or the person in charge would add that product to the cart and after selling this product the automated inventory will substract the product from the stock automatically. After completing the product selling printed receipt will be given to the customer and sur price will be provided after completing the deduction of TAX and VAT. The whole process is below illustrated according to the algorithm for selling 1 bunch of Banana.

Step 1: Start of the application

Step 2: Check the stock Go Master Entry.

Step 3: Product>New>insert Product Name(if not exist) for example :-Banana

Step 4: If yes: insert product name(Banana), product category(Fruit),sub category(summer fruit), Supplier(KMS) with description of amount(10000 bunch)received . and select the customer ID (If exists) to give discount on purchase .

Step 5: If No: Click close

Step 6: if yes open the Stock In where product price *(Minimum and Maximum will be inserted manually) quantity(Generate from supplier) and – amount received

Step 7: Click save and add this to the inventory for trasactiomn

Step 9: From Billing section listed product generated from STOCK in>Select BANANA>select quantity>Add to card for sale>CLICK save button

Step 10: If Yes: Proceed to print

Step 11: If No: Close

Step 12: IF Decide to erase it >Press delete button

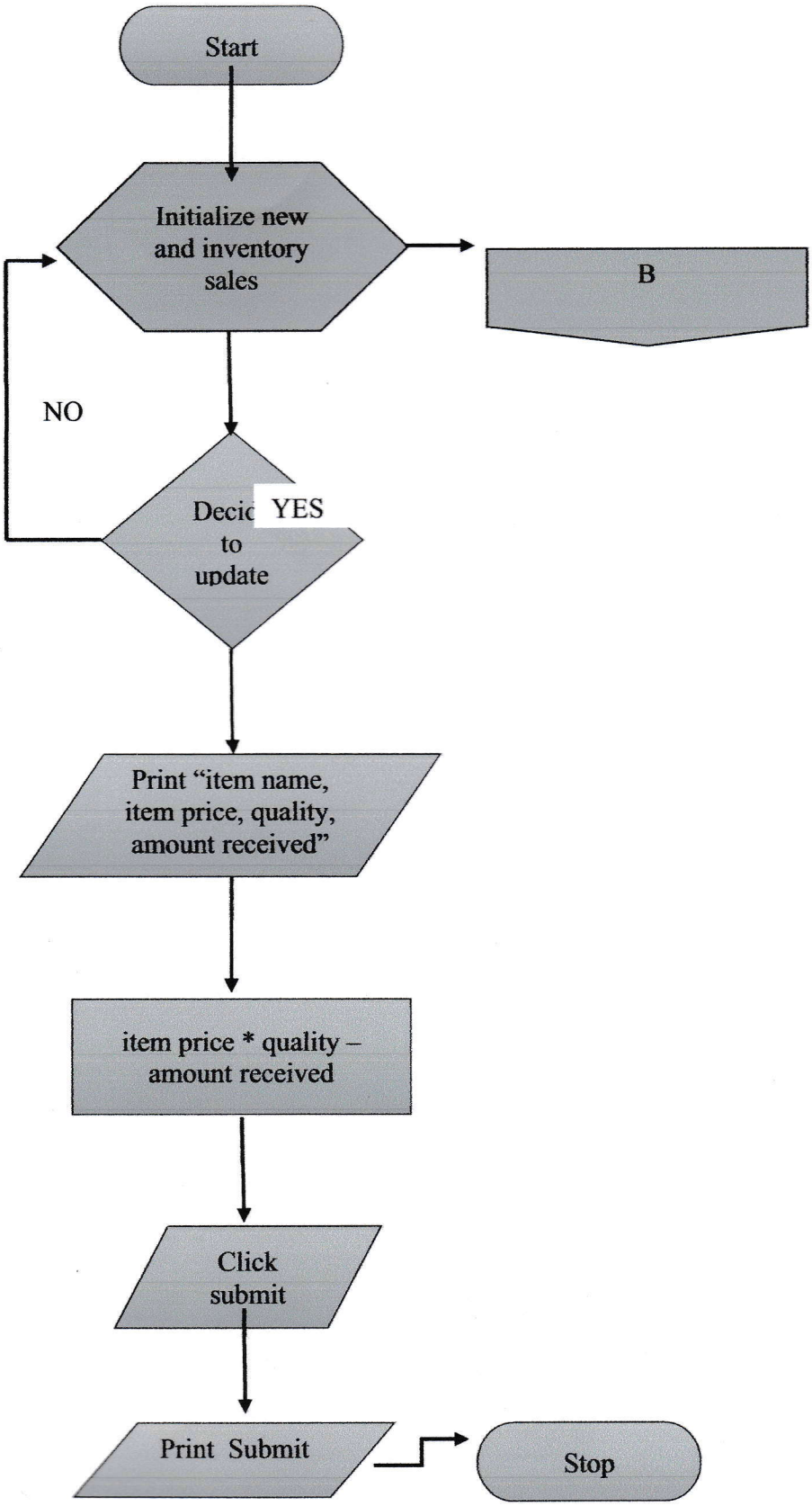
Step 13: If Yes:(For successful transaction) Print receipt (with vat and discount !!!")

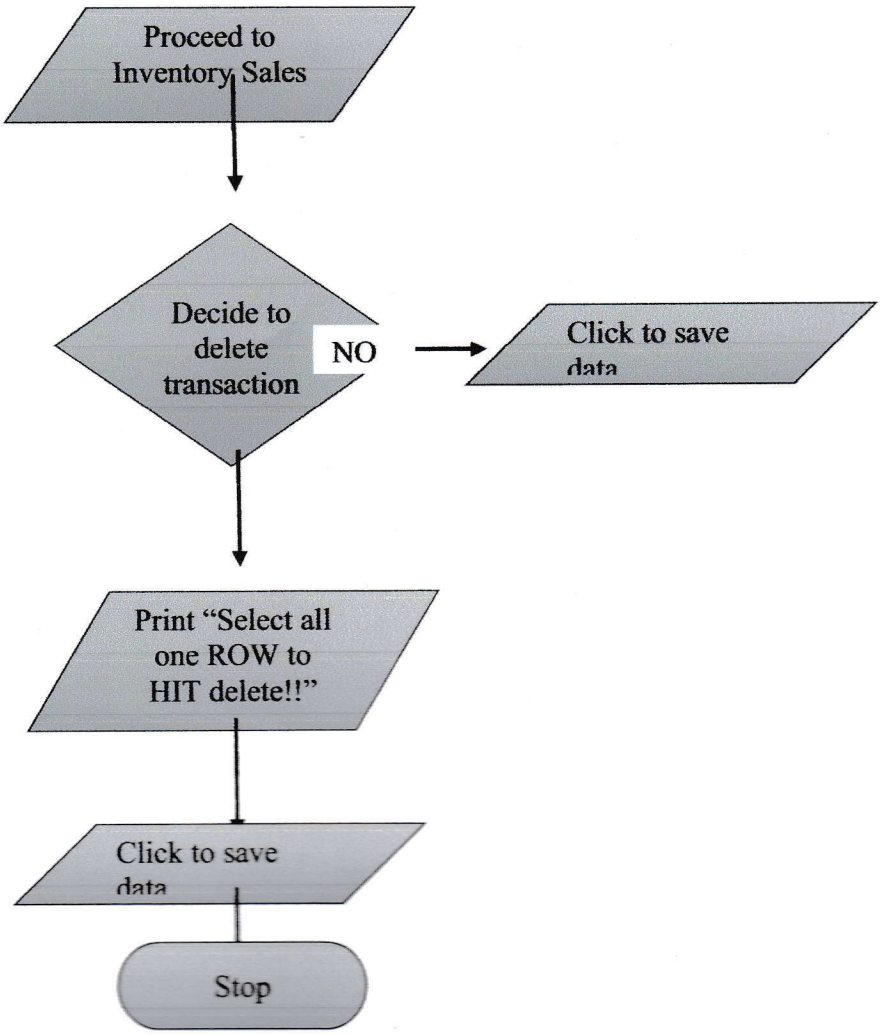
Step 14: If No: Click to save data (for update transaction infuture)

Step 15: Stop the program.

4.3 Flowchart

The flowchart in adding a transaction.



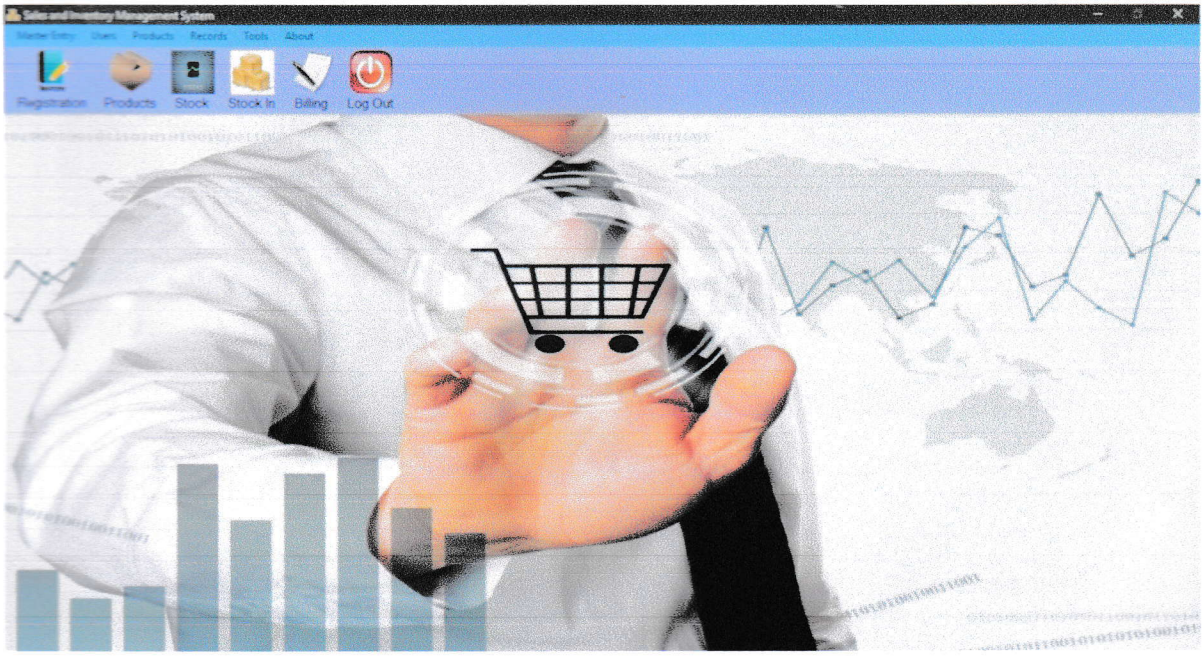


4.4. *Front End Design*



Fig :Front end design

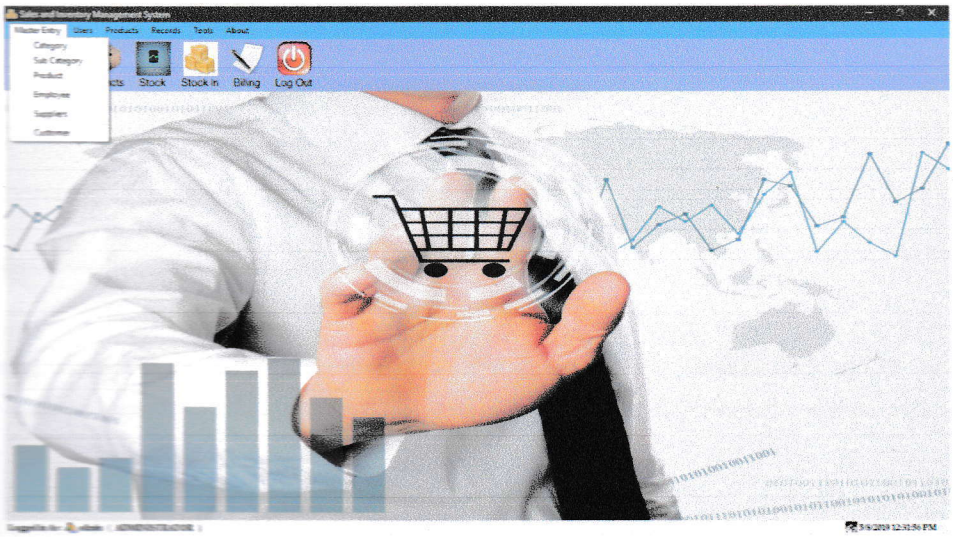
4.5User end interface



Executions : - After login interface which consists all the navbar,menubar and panel for executions .In this interaface there are 3 types of executions will be held in order.They are ordered below:-

1. Entry product By their category
2. Product in from the supplier to stock
3. Stock to Invetory for Billing

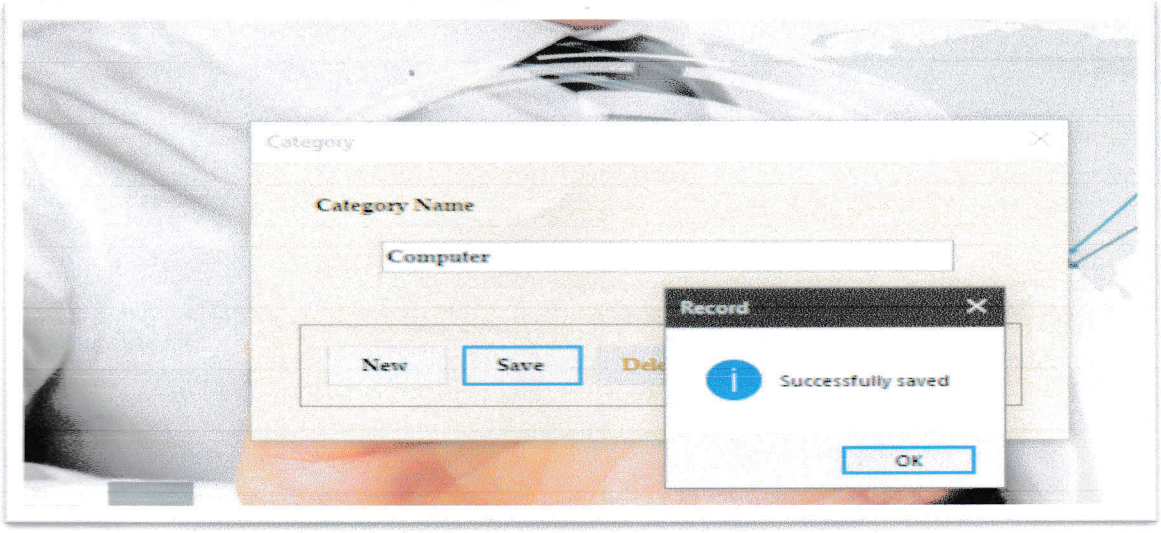
4.6 Enter the Product



From master entry we can enter the Product order by

After inserting any category or any subcategory and product inside the inventory we can check it through the button **get data.**

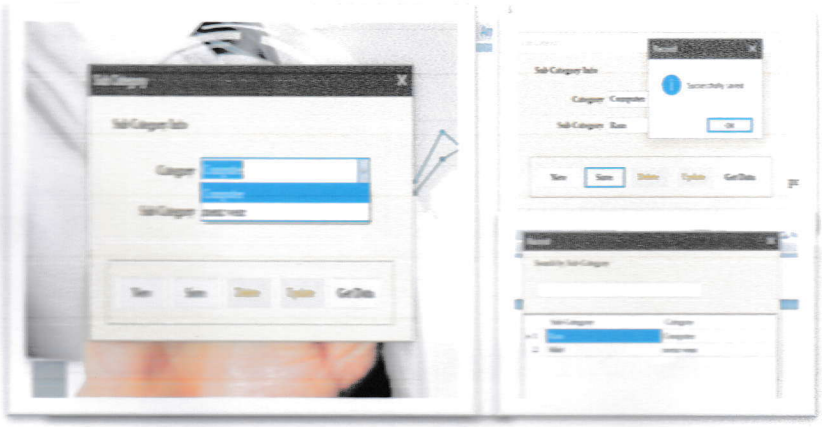
4.7Enter Category



Inserted category will be checked after pressing get data and category name will fair to any product relevant name.



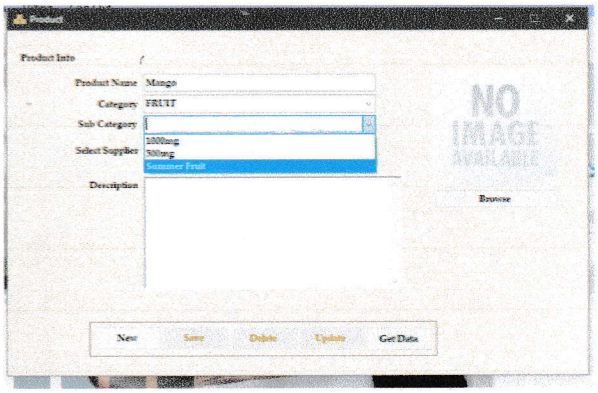
4.8Enter Subcategory



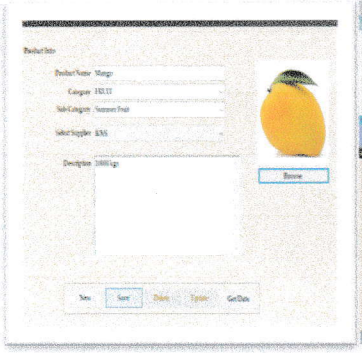
After entering category it will go to the database and under this category as ancestor,there will be unlimited subcategory can be created.Like this , all the products that comes from the direct supplier will be under any sub category

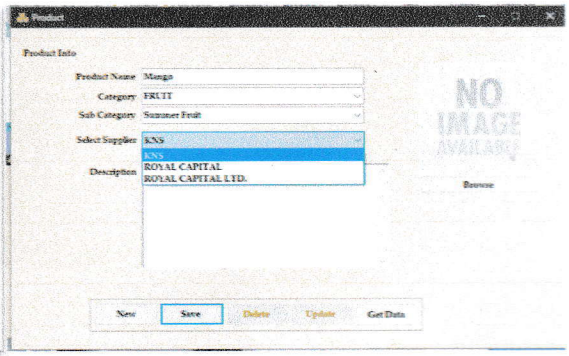
4.9Enter Product

To select product _after passing category>Sub-category and after select the supplier we have insert the amount got from the supplier exactly.This is predined that we have to select the product name and the amount in the description.Image of the relevant product can be uploaded as well.

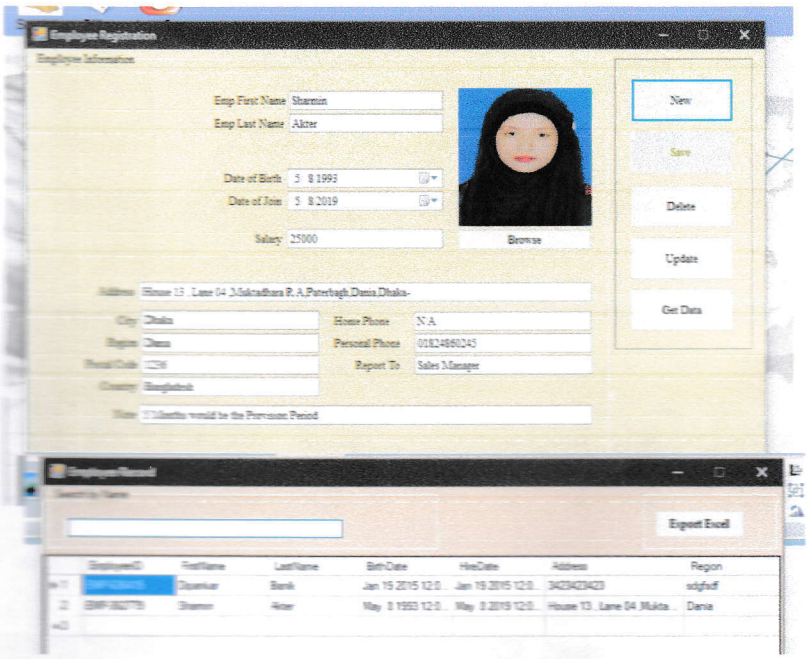






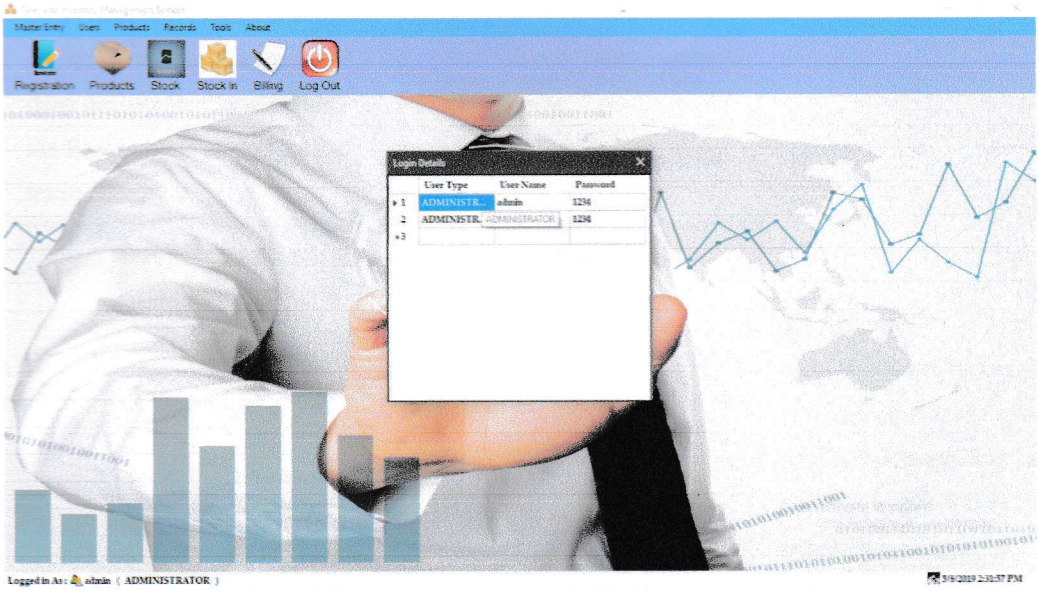


5.0Enter Employee Resigtration

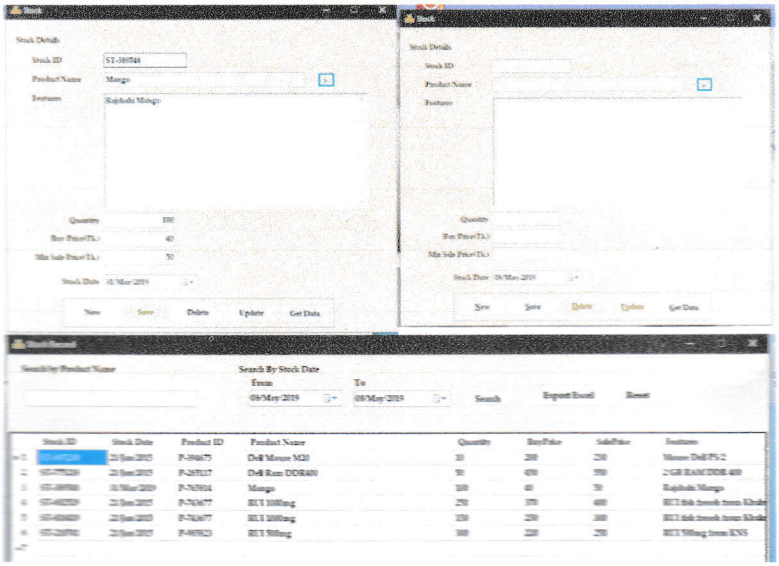


From master entry all the employee resigtration could be completed by seleting employee title .Some mandatory information will be needed as required.And eventuallly after saving all information ,all can be rerieved from get data as repvious.

In login deatails,an admin can see the details of activities of any employees in the store and even the password as well.An admin is permitted to edit and delete any user froim this panel



5.2 Stock in for inventory

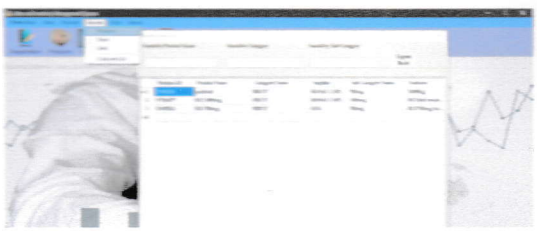


Stock in inventory gets data from the initial entryn of products from supplier.From stock in products goes for inventory or RAW stock for final transaction inside the business store.

5.3 Record Panel

In record panel there are 4 types of data can be retrieved.

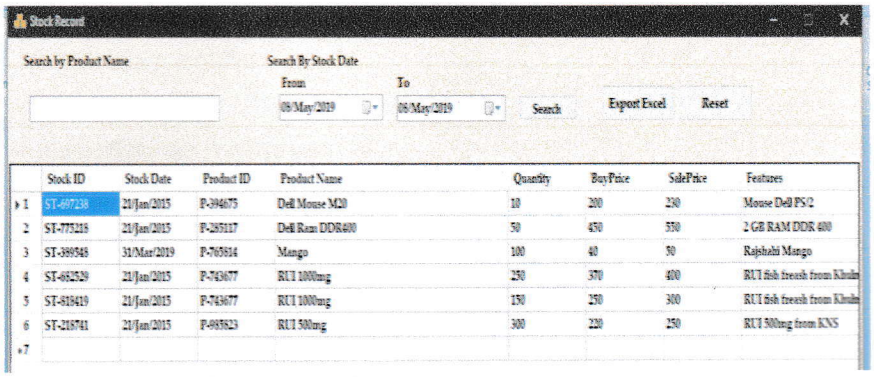
5.3.1.products

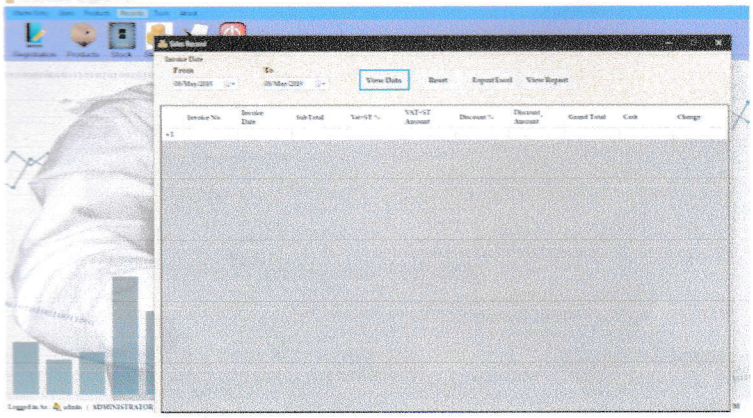


Products panel is listed for inventory or final stock in for the transaction.Actually this is the panel where products stays for final approval products from the supplier.

5.3.2 Stock

Stocks are the platform whrere suppliers are intend to store their product for final approval from the admin.Aftter select and giving the amount that received from the supplier ,save it.How the final stock is ready for final transaction.



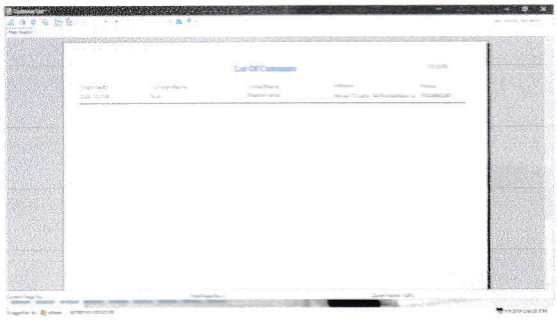


5.3.3. Sales

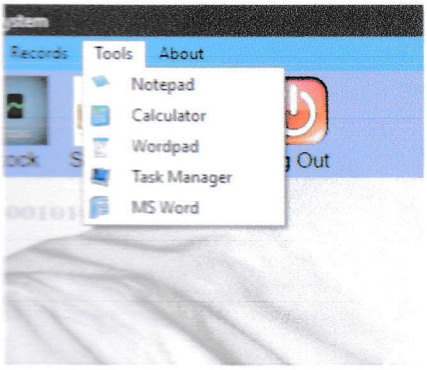
Sales list or the successful trasaction list that happened in the store.After a successful transaction and save this trasaction only will be displayed in this panel.This panel is only for the logs of records.

5.3.4 Customer Records

Successful resigtration of any customers from the main panel entry is followed up here

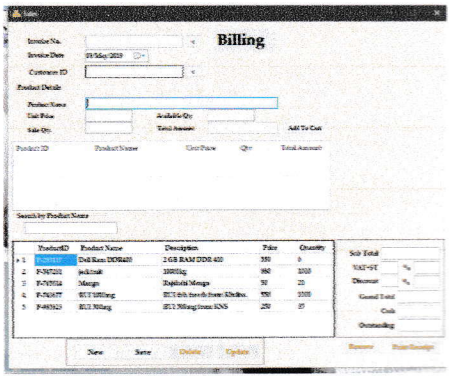


5.4 Additional Tools



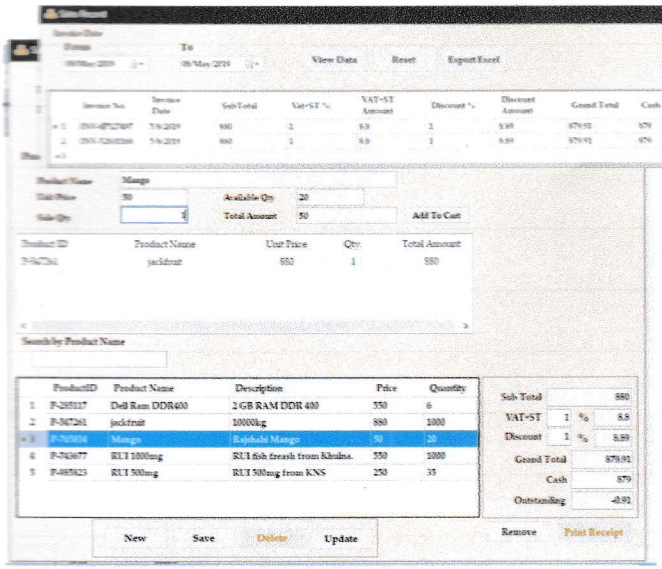
Additional tools are integrated additionally fotr user interface and helping and make it friendly.they are Notepad,Calculator,MS word,Task Manager ,Wordpad Shorcuts .

5.5 Billing



Billing is the final panmel where a transactions are fulfilled to go and invoices are generated .

In this phase to go for product transaction,from the product panel of billing section, follow this procedureto go for a Successful transaction.



Could possible to find out generated invoice from get data button of billing

Select product>Enter Quantity>Add this transaction to the cart>Click save and insert the required VAT and Discount according to the customer.Save and print receipt or Invoice

CHAPTER 05

Source Codes



5.1 For Section Category

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;
namespace Sales_and_Inventory_System
{
  public partial class frmCategory : Form
  {
    SqlDataReader rdr = null;
    DataTable dtable = new DataTable();
    SqlConnection con = null;
    SqlCommand cmd = null;
    DataTable dt = new DataTable();
    ConnectionString cs = new ConnectionString();


    public frmCategory()
    {
      InitializeComponent();
    }

    private void frmCategory_Load(object sender, EventArgs e)
    {
      btnSave.Enabled = false;
      Autocomplete();
    }

    private void btnSave_Click(object sender, EventArgs e)
    {
      if (txtCategoryName.Text == "")
      {
        MessageBox.Show("Please enter Category name", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        txtCategoryName.Focus();
        return;
      }


      try
      {

        con = new SqlConnection(cs.DBConn);
        con.Open();
        string ct = "SELECT CategoryName from T_Category where CategoryName='" + txtCategoryName.Text
+ "'";

        cmd = new SqlCommand(ct);
        cmd.Connection = con;
        rdr = cmd.ExecuteReader();

        if (rdr.Read())
        {
          MessageBox.Show("Category Name Already Exists", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
          txtCategoryName.Text = "";
          txtCategoryName.Focus();
```

```
            {
                rdr.Close();
            }
            return;
        }

        con = new SqlConnection(cs.DBConn);
        con.Open();

        string cb = "INSERT INTO T_Category(CategoryName) VALUES ('" + txtCategoryName.Text + "')";

        cmd = new SqlCommand(cb);
        cmd.Connection = con;
        cmd.ExecuteReader();
        con.Close();
        MessageBox.Show("Successfully saved", "Record", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        Autocomplete();
        btnSave.Enabled = false;

      }
      catch (Exception ex)
      {
        MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
      }
    }

    private void btnDelete_Click(object sender, EventArgs e)
    {
      if (MessageBox.Show("Do you really want to delete this record?", "Confirmation",
MessageBoxButtons.YesNo, MessageBoxIcon.Question) == System.Windows.Forms.DialogResult.Yes)
      {
        delete_records();
      }
    }
    private void delete_records()
    {

      try
      {

        int RowsAffected = 0;
        con = new SqlConnection(cs.DBConn);
        con.Open();
        string cq = "DELETE FROM T_Category where ID=" + txtID.Text + "";
        cmd = new SqlCommand(cq);
        cmd.Connection = con;
        RowsAffected = cmd.ExecuteNonQuery();
        if (RowsAffected > 0)
        {
          MessageBox.Show("Successfully deleted", "Record", MessageBoxButtons.OK,
MessageBoxIcon.Information);
          Reset();
          Autocomplete();
        }
        else
        {
          MessageBox.Show("No Record found", "Sorry", MessageBoxButtons.OK,
MessageBoxIcon.Information);
          Reset();
          Autocomplete();
        }
        if (con.State == ConnectionState.Open)
        {
```

```
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
    private void Autocomplete()
    {
        try
        {
            con = new SqlConnection(cs.DBConn);
            con.Open();
            SqlCommand cmd = new SqlCommand("SELECT distinct Categoryname FROM T_Category", con);
            DataSet ds = new DataSet();
            SqlDataAdapter da = new SqlDataAdapter(cmd);
            da.Fill(ds, "T_Category");
            AutoCompleteStringCollection col = new AutoCompleteStringCollection();
            int i = 0;
            for (i = 0; i <= ds.Tables[0].Rows.Count - 1; i++)
            {
                col.Add(ds.Tables[0].Rows[i]["Categoryname"].ToString());

            }
            txtCategoryName.AutoCompleteSource = AutoCompleteSource.CustomSource;
            txtCategoryName.AutoCompleteCustomSource = col;
            txtCategoryName.AutoCompleteMode = AutoCompleteMode.Suggest;

            con.Close();
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
    private void btnUpdate_Click(object sender, EventArgs e)
    {
        try
        {
            if (txtCategoryName.Text == "")
            {
                MessageBox.Show("Please enter Category name", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                txtCategoryName.Focus();
                return;
            }
            con = new SqlConnection(cs.DBConn);
            con.Open();

            string cb = "UPDATE T_Category set CategoryName='" + txtCategoryName.Text + "' where ID=" +
txtID.Text + "";
            cmd = new SqlCommand(cb);
            cmd.Connection = con;
            cmd.ExecuteReader();
            con.Close();
            MessageBox.Show("Successfully updated", "Record", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            Autocomplete();
            btnUpdate.Enabled = false;
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
```

```
        btnSave.Enabled = true;
        btnDelete.Enabled = false;
        btnUpdate.Enabled = false;
        txtCategoryName.Focus();
    }

    private void btnNew_Click(object sender, EventArgs e)
    {
        Reset();
    }

    private void btnGetData_Click(object sender, EventArgs e)
    {
        this.Hide();
        frmCategoryRecord frm = new frmCategoryRecord();
        frm.Show();
        frm.GetData();
    }
  }
}
```

5.2For category record

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Sales_and_Inventory_System
{
    public partial class frmCategoryRecord : Form
    {
        SqlDataReader rdr = null;
        SqlConnection con = null;
        SqlCommand cmd = null;
        ConnectionString cs = new ConnectionString();
        public frmCategoryRecord()
        {
            InitializeComponent();
        }

        private void dataGridView1_RowPostPaint(object sender,
DataGridViewRowPostPaintEventArgs e)
        {
            string strRowNumber = (e.RowIndex + 1).ToString();
            SizeF size = e.Graphics.MeasureString(strRowNumber, this.Font);
            if (dataGridView1.RowHeadersWidth < Convert.ToInt32((size.Width + 20)))
            {
                dataGridView1.RowHeadersWidth = Convert.ToInt32((size.Width + 20));
            }
            Brush b = SystemBrushes.ControlText;
            e.Graphics.DrawString(strRowNumber, this.Font, b, e.RowBounds.Location.X +
15, e.RowBounds.Location.Y + ((e.RowBounds.Height - size.Height) / 2));

        }

        private void dataGridView1_RowHeaderMouseClick(object sender,
DataGridViewCellMouseEventArgs e)
        {
            DataGridViewRow dr = dataGridView1.SelectedRows[0];
            this.Hide();
```

```
            frm.txtCategoryName.Text = dr.Cells[1].Value.ToString();
            frm.btnDelete.Enabled = true;
            frm.btnUpdate.Enabled = true;
            frm.txtCategoryName.Focus();
            frm.btnSave.Enabled = false;
        }
        public void GetData()
        {
            try
            {
                con = new SqlConnection(cs.DBConn);
                con.Open();
                String sql = "SELECT * FROM T_Category order by CategoryName";
                cmd = new SqlCommand(sql, con);
                rdr = cmd.ExecuteReader(CommandBehavior.CloseConnection);
                dataGridView1.Rows.Clear();
                while (rdr.Read() == true)
                {
                    dataGridView1.Rows.Add(rdr[0],rdr[1]);
                }
                con.Close();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        private void frmCategoryRecord_FormClosing(object sender, FormClosingEventArgs e)
        {
            this.Hide();
            frmCategory frm = new frmCategory();
            frm.Show();
        }

        private void frmCategoryRecord_Load(object sender, EventArgs e)
        {
            GetData();
        }
    }
```

5.3Registration Panel

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.IO;
using System.Security.Cryptography;


namespace Sales and Inventory System
```

```
SqlConnection con = null;
SqlCommand cmd = null;
DataTable dt = new DataTable();
ConnectionString cs = new ConnectionString();
public frmRegistration()
{
   InitializeComponent();
}

private void Form1_Load(object sender, EventArgs e)
{
   btnRegister.Enabled = false;
   Autocomplete();
}
private void Reset()
{
   txtUsername.Text = "";
   cmbUserType.SelectedIndex = -1;
   txtPassword.Text = "";
   txtContact_no.Text = "";
   txtName.Text = "";
   txtEmail_Address.Text = "";

   btnRegister.Enabled = true;
   btnDelete.Enabled = false;
   btnUpdate_record.Enabled = false;
   txtUsername.Focus();
}
private void NewRecord_Click(object sender, EventArgs e)
{
   Reset();
}

private void Register_Click(object sender, EventArgs e)
{
   if (txtUsername.Text == "")
   {
      MessageBox.Show("Please enter username", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
      txtUsername.Focus();
      return;
   }
   if (cmbUserType.Text == "")
   {
      MessageBox.Show("Please select user type", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
      cmbUserType.Focus();
      return;
   }
   if (txtPassword.Text == "")
   {
      MessageBox.Show("Please enter password", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
      txtPassword.Focus();
      return;
   }
   if (txtName.Text == "")
   {
      MessageBox.Show("Please enter name", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
      txtName.Focus();
      return;
   }
   if (txtContact_no.Text == "")
   {
      MessageBox.Show("Please enter contact no.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
      txtContact_no.Focus();
```

```
            txtEmail_Address.Focus();
            return;
        }
        try
        {
            con = new SqlConnection(cs.DBConn);
            con.Open();
            string ct = "SELECT username from T_REGISTRATION where username=@find";

            cmd = new SqlCommand(ct);
            cmd.Connection = con;
            cmd.Parameters.Add(new SqlParameter("@find", System.Data.SqlDbType.NChar, 30, "username"));
            cmd.Parameters["@find"].Value = txtUsername.Text;
            rdr = cmd.ExecuteReader();

            if (rdr.Read())
            {
                MessageBox.Show("Username Already Exists", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                txtUsername.Text = "";
                txtUsername.Focus();


                if ((rdr != null))
                {
                    rdr.Close();
                }
                return;
            }
            con = new SqlConnection(cs.DBConn);
            con.Open();
            string ct1 = "SELECT Email from T_REGISTRATION where Email='" + txtEmail_Address.Text + "'";

            cmd = new SqlCommand(ct1);
            cmd.Connection = con;
            rdr = cmd.ExecuteReader();

            if (rdr.Read())
            {
                MessageBox.Show("Email ID Already Exists", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                txtEmail_Address.Text = "";
                txtEmail_Address.Focus();


                if ((rdr != null))
                {
                    rdr.Close();
                }
                return;
            }
            con = new SqlConnection(cs.DBConn);
            con.Open();

            string cb = "INSERT INTO
T_REGISTRATION(Username,UserType,Password,ContactNo,Email,Name,JoiningDate) VALUES ('" +
txtUsername.Text + "','" + cmbUserType.Text + "','" + txtPassword.Text + "','" + txtContact_no.Text + "','" +
txtEmail_Address.Text + "','" + txtName.Text + "','" + System.DateTime.Now + "')";


    cmd = new SqlCommand(cb);
            cmd.Connection = con;
```

```
            btnRegister.Enabled = false;

        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }


    private void CheckAvailability_Click(object sender, EventArgs e)
    {
        try
        {
            if (txtUsername.Text == "")
            {
                MessageBox.Show("Please enter username", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                txtUsername.Focus();
                return;
            }
            con = new SqlConnection(cs.DBConn);
            con.Open();
            string ct = "SELECT username from T_REGISTRATION where username='" + txtUsername.Text + "'";

            cmd = new SqlCommand(ct);
            cmd.Connection = con;
            rdr = cmd.ExecuteReader();

            if (rdr.Read())
            {
                MessageBox.Show("Username not available", "", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                return;
            }
            if (!rdr.Read())
            {
                MessageBox.Show("Username available", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
                txtUsername.Focus();

            }
            if ((rdr != null))
            {
                rdr.Close();
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

    private void Email_Address_Validating(object sender, CancelEventArgs e)
    {
        System.Text.RegularExpressions.Regex rEMail = new System.Text.RegularExpressions.Regex(@"^[a-zA-
Z][\w\.-]{2,28}[a-zA-Z0-9]@[a-zA-Z0-9][\w\.-]*[a-zA-Z0-9]\.[a-zA-Z][a-zA-Z\.]*[a-zA-Z]$");
        if (txtEmail_Address.Text.Length > 0)
        {
            if (!rEMail.IsMatch(txtEmail_Address.Text))
            {
                MessageBox.Show("invalid email address", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                txtEmail_Address.SelectAll();
                e.Cancel = true;
```

```
    {
      e.Handled = !(char.IsLetter(e.KeyChar) || e.KeyChar == (char)Keys.Back || e.KeyChar ==
(char)Keys.Space);
    }

    private void Username_Validating(object sender, CancelEventArgs e)
    {
      System.Text.RegularExpressions.Regex rEMail = new System.Text.RegularExpressions.Regex("^[a-zA-Z0-
9_]");
      if (txtUsername.Text.Length > 0)
      {
        if (!rEMail.IsMatch(txtUsername.Text))
        {
          MessageBox.Show("only letters,numbers and underscore is allowed", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
          txtUsername.SelectAll();
          e.Cancel = true;
        }
      }
    }

    private void GetDetails_Click(object sender, EventArgs e)
    {
      frmRegisteredUsersDetails frm = new frmRegisteredUsersDetails();
      frm.dataGridView1.DataSource = frm.GetData();
      frm.Show();

    }

    private void Username_TextChanged(object sender, EventArgs e)
    {

      btnDelete.Enabled = true;
      btnUpdate_record.Enabled = true;
      try
      {
        txtUsername.Text = txtUsername.Text.TrimEnd();
        con = new SqlConnection(cs.DBConn);

        con.Open();
        cmd = con.CreateCommand();

        cmd.CommandText = "SELECT Usertype,password,name,contactno,email FROM T_REGISTRATION
WHERE username = '" + txtUsername.Text.Trim() + "'";
        rdr = cmd.ExecuteReader();

        if (rdr.Read())
        {
          cmbUserType.Text = (rdr.GetString(0).Trim());
          txtPassword.Text = (rdr.GetString(1).Trim());
          txtName.Text = (rdr.GetString(2).Trim());
          txtContact_no.Text = (rdr.GetString(3).Trim());
          txtEmail_Address.Text = (rdr.GetString(4).Trim());
        }

        if ((rdr != null))
        {
          rdr.Close();
        }
        if (con.State == ConnectionState.Open)
        {
          con.Close();
        }
```

```
            MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

    private void Autocomplete()
    {
        try{
        con = new SqlConnection(cs.DBConn);
        con.Open();
        SqlCommand cmd = new SqlCommand("SELECT UserName FROM T_REGISTRATION", con);
        DataSet ds = new DataSet();
        SqlDataAdapter da = new SqlDataAdapter(cmd);
        da.Fill(ds, "T_REGISTRATION");
        AutoCompleteStringCollection col = new AutoCompleteStringCollection();
        int i = 0;
        for (i = 0; i <= ds.Tables[0].Rows.Count - 1; i++)
        {
            col.Add(ds.Tables[0].Rows[i]["UserName"].ToString());

        }
        txtUsername.AutoCompleteSource = AutoCompleteSource.CustomSource;
        txtUsername.AutoCompleteCustomSource = col;
        txtUsername.AutoCompleteMode = AutoCompleteMode.Suggest;

        con.Close();
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

    private void Update_record_Click(object sender, EventArgs e)
    {
        if (txtUsername.Text == "")
        {
            MessageBox.Show("Please enter UserID", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            txtUsername.Focus();
            return;
        }
        if (cmbUserType.Text == "")
        {
            MessageBox.Show("Please select User Type", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            cmbUserType.Focus();
            return;
        }
        if (txtPassword.Text == "")
        {
            MessageBox.Show("Please enter Password", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            txtPassword.Focus();
            return;
        }
        if (txtName.Text == "")
        {
            MessageBox.Show("Please enter User Name", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            txtName.Focus();
            return;
        }
        if (txtContact_no.Text == "")
        {
            MessageBox.Show("Please enter contact no.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            txtContact_no.Focus();
            return;
        }
```

```
            return;
        }
        try
        {
            con = new SqlConnection(cs.DBConn);
            con.Open();

            string cb = "UPDATE T_REGISTRATION SET UserType='" + cmbUserType.Text + "', Password='" +
txtPassword.Text + "',ContactNo='" + txtContact_no.Text + "',Email='" + txtEmail_Address.Text + "',Name='" +
txtName.Text + "' WHERE UserName='" + txtUsername.Text + "'";
            cmd = new SqlCommand(cb);
            cmd.Connection = con;
            cmd.ExecuteReader();
            con.Close();

            MessageBox.Show("Successfully updated", "User Info", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            Autocomplete();
            btnUpdate_record.Enabled = false;
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

    private void Delete_Click(object sender, EventArgs e)
    {
        if (MessageBox.Show("Do you really want to delete this record?", "Confirmation",
MessageBoxButtons.YesNo, MessageBoxIcon.Question) == System.Windows.Forms.DialogResult.Yes)
        {
            delete_records();
        }
    }
    private void delete_records()
    {

        try
        {
            if (txtUsername.Text == "admin")
            {
                MessageBox.Show("Admin Account can not be deleted", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                return;
            }
            int RowsAffected = 0;
            con = new SqlConnection(cs.DBConn);
            con.Open();
            string ct = "DELETE FROM T_REGISTRATION WHERE UserName='" + txtUsername.Text + "'";
            cmd = new SqlCommand(ct);
            cmd.Connection = con;
            RowsAffected = cmd.ExecuteNonQuery();
            if (RowsAffected > 0)
            {
                MessageBox.Show("Successfully deleted", "Record", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                Autocomplete();
                Reset();
            }
            else
            {
                MessageBox.Show("No Record found", "Sorry", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                Reset();
```

```
            con.Close();
        }

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}


private void txtContact_no_KeyPress(object sender, KeyPressEventArgs e)
{
    if (char.IsDigit(e.KeyChar) || char.IsControl(e.KeyChar))
    {
        e.Handled = false;
    }
    else
    {
        e.Handled = true;
    }
}
```

5.4For Main Menu

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Data.SqlClient;


namespace Sales_and_Inventory_System
{
    public partial class frmMainMenu : Form
    {
        ConnectionString cs = new ConnectionString();
        public frmMainMenu()
        {


            InitializeComponent();
        }



        private void registrationToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            frmRegistration frm = new frmRegistration();
            frm.Show();
        }

        private void aboutToolStripMenuItem_Click(object sender, EventArgs e)
```

```
private void registrationToolStripMenuItem_Click(object sender, EventArgs e)
{
    frmRegistration frm = new frmRegistration();
    frm.Show();
}

private void productToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (lblUserType.Text == "ADMINISTRATOR")
    {
        frmProduct frm = new frmProduct();
        frm.lblUserType.Text = lblUserType.Text;
        frm.btnGetData.Enabled = true;
        frm.Show();
    }
    if (lblUserType.Text == "INVENTORY MANAGER")
    {
        frmProduct frm = new frmProduct();
        frm.lblUserType.Text = lblUserType.Text;
        frm.btnGetData.Enabled = false;
        frm.Show();
    }
}

private void notepadToolStripMenuItem_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process.Start("Notepad.exe");
}

private void calculatorToolStripMenuItem_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process.Start("Calc.exe");
}

private void wordpadToolStripMenuItem_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process.Start("Wordpad.exe");
}

private void taskManagerToolStripMenuItem_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process.Start("TaskMgr.exe");
}

private void mSWordToolStripMenuItem_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process.Start("Winword.exe");
}

private void categoryToolStripMenuItem_Click(object sender, EventArgs e)
{
    frmCategory frm = new frmCategory();
    frm.Show();
}

private void companyToolStripMenuItem_Click(object sender, EventArgs e)
{
    frmSubCategory frm = new frmSubCategory();
    frm.Show();
}
```

```
        //frmSubCategory o2 = new frmSubCategory();
        //o2.Hide();
        //frmProduct o3 = new frmProduct();
        //o3.Hide();
        //frmRegisteredUsersDetails o4 = new frmRegisteredUsersDetails();
        //o4.Hide();
        //frmRegistration o5 = new frmRegistration();
        //o5.Hide();
        //frmProductsRecord2 o9 = new frmProductsRecord2();
        //o9.Hide();
        //frmSalesRecord o10 = new frmSalesRecord();
        //o10.Hide();
        //frmStockIN o11=new frmStockIN();
        //o11.Hide();

        this.Dispose();

        frmLogin frm = new frmLogin();
        //frm.cmbUsertype.SelectedIndex = -1;
        //frm.txtUserName.Text = "";
        //frm.txtPassword.Text = "";
        //frm.ProgressBar1.Visible = false;
        //frm.cmbUsertype.Focus();
        frm.Show();
    }


    private void frmMainMenu_Load(object sender, EventArgs e)
    {
        lblTime.Text = System.DateTime.Now.ToString();

    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        lblTime.Text = System.DateTime.Now.ToString();
    }

    private void productsToolStripMenuItem1_Click(object sender, EventArgs e)
    {
        if (lblUserType.Text == "ADMINISTRATOR")
        {
            frmProduct frm = new frmProduct();
            frm.lblUserType.Text = lblUserType.Text;
            frm.btnGetData.Enabled = true;
            frm.Show();
        }
        if (lblUserType.Text == "INVENTORY MANAGER")
        {
            frmProduct frm = new frmProduct();
            frm.lblUserType.Text = lblUserType.Text;
            frm.btnGetData.Enabled = false;
            frm.Show();
        }
    }

    private void productsToolStripMenuItem2_Click(object sender, EventArgs e)
    {
        frmProductsRecord2 frm = new frmProductsRecord2();
        frm.Show();
    }
```

```
            frmStock frm = new frmStock();
            frm.lblUserType.Text = lblUserType.Text;
            frm.btnGetData.Enabled = true;
            frm.Show();
        }
        if (lblUserType.Text == "INVENTORY MANAGER")
        {
            frmStock frm = new frmStock();
            frm.lblUserType.Text = lblUserType.Text;
            frm.btnGetData.Enabled = false;
            frm.Show();
        }
    }

    private void stockToolStripMenuItem_Click(object sender, EventArgs e)
    {
        if (lblUserType.Text == "ADMINISTRATOR")
        {
            frmStock frm = new frmStock();
            frm.lblUserType.Text = lblUserType.Text;
            frm.btnGetData.Enabled = true;
            frm.Show();
        }
        if (lblUserType.Text == "INVENTORY MANAGER")
        {
            frmStock frm = new frmStock();
            frm.lblUserType.Text = lblUserType.Text;
            frm.btnGetData.Enabled = false;
            frm.Show();
        }
    }



    private void invoiceToolStripMenuItem_Click(object sender, EventArgs e)
    {
        if (lblUserType.Text == "ADMINISTRATOR")
        {
            frmSales frm = new frmSales();
            frm.txtUserID.Text = this.lbluserid.Text;
            frm.lblUserType.Text = lblUserType.Text;
            frm.btnGetData.Enabled = true;
            frm.Show();
        }
        if (lblUserType.Text == "SALES PERSON")
        {
            frmSales frm = new frmSales();
            frm.txtUserID.Text = this.lbluserid.Text;
            frm.lblUserType.Text = lblUserType.Text;
            frm.btnGetData.Enabled = false;
            frm.Show();
        }
    }

    private void salesToolStripMenuItem_Click(object sender, EventArgs e)
    {
        if (lblUserType.Text == "ADMINISTRATOR")
        {
            frmSales frm = new frmSales();
            frm.lblUserType.Text = lblUserType.Text;
            frm.btnGetData.Enabled = true;
```

```
            frm.lblUserType.Text = lblUserType.Text;
            frm.btnGetData.Enabled = false;
            frm.Show();
        }
    }

    private void salesToolStripMenuItem1_Click(object sender, EventArgs e)
    {
        frmSalesRecord frm = new frmSalesRecord();
        frm.Show();
    }

    private void loginDetailsToolStripMenuItem_Click(object sender, EventArgs e)
    {
        frmLoginDetails frm = new frmLoginDetails();
        frm.Show();
    }


    private void frmMainMenu_FormClosing(object sender, FormClosingEventArgs e)
    {
       // this.Dispose();
        e.Cancel = true;


    }



    private void stockToolStripMenuItem2_Click(object sender, EventArgs e)
    {
        frmStockRecord1 frm = new frmStockRecord1();
        frm.Show();
    }

    private void stockInToolStripMenuItem_Click(object sender, EventArgs e)
    {
        frmStockIN frm = new frmStockIN();
        frm.Show();
    }

    private void frmMainMenu_FormClosed(object sender, FormClosedEventArgs e)
    {

    }

    private void employeeToolStripMenuItem_Click(object sender, EventArgs e)
    {
        if (lblUserType.Text == "ADMINISTRATOR")
        {
            frmEmp frm = new frmEmp();
            frm.lblUserType.Text = lblUserType.Text;
            frm.btnGetData.Enabled = true;
            frm.Show();
        }
    }

    private void suppliersToolStripMenuItem_Click(object sender, EventArgs e)
    {
        if (lblUserType.Text == "ADMINISTRATOR")
        {
            frmSuppliers frm = new frmSuppliers();
```

```
            {
                frmSuppliers frm = new frmSuppliers();
                frm.lblUserType.Text = lblUserType.Text;
                frm.btnGetData.Enabled = false;
                frm.Show();
            }
        }

        private void customerToolStripMenuItem_Click(object sender, EventArgs e)
        {

            if (lblUserType.Text == "ADMINISTRATOR")
            {
                frmCustomer frm = new frmCustomer();
                frm.lblUserType.Text = lblUserType.Text;
                frm.btnGetData.Enabled = true;
                frm.Show();
            }
            if (lblUserType.Text == "INVENTORY MANAGER")
            {
                frmCustomer frm = new frmCustomer();
                frm.lblUserType.Text = lblUserType.Text;
                frm.btnGetData.Enabled = false;
                frm.Show();
            }

        }

        private void customerToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            try
            {

                CrystalReportCustomerDetails rpt = new
CrystalReportCustomerDetails();

                SqlCommand cmd = new SqlCommand();
                SqlDataAdapter myDA = new SqlDataAdapter();
                POS_DBDataSet myDS = new POS_DBDataSet();
                //The DataSet you created.
                SqlConnection con = new SqlConnection(cs.DBConn);
                cmd.Connection = con;
                cmd.CommandText = "SELECT * from T_CUSTOMER";
                cmd.CommandType = CommandType.Text;
                myDA.SelectCommand = cmd;
                myDA.Fill(myDS, "T_CUSTOMER");
                rpt.SetDataSource(myDS);

                frmCustomerListRecord frm = new frmCustomerListRecord();
                frm.crystalReportViewer1.ReportSource = rpt;
                frm.Visible = true;
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

    ////////////////////////////////
    }
}
```

5.5For Product Entry

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.IO;
using System.Security.Cryptography;

namespace Sales_and_Inventory_System
{
    public partial class frmProduct : Form
    {
        SqlDataReader rdr = null;
        DataTable dtable = new DataTable();
        SqlConnection con = null;
        SqlCommand cmd = null;
        DataTable dt = new DataTable();
       ConnectionString cs = new ConnectionString();
        public frmProduct()
        {
            InitializeComponent();
        }
        private void auto()
        {
            txtProductID.Text = "P-" + GetUniqueKey(6);
        }
        public static string GetUniqueKey(int maxSize)
        {
            char[] chars = new char[62];
            chars = "123456789".ToCharArray();
            byte[] data = new byte[1];
            RNGCryptoServiceProvider crypto = new RNGCryptoServiceProvider();
            crypto.GetNonZeroBytes(data);
            data = new byte[maxSize];
            crypto.GetNonZeroBytes(data);
            StringBuilder result = new StringBuilder(maxSize);
            foreach (byte b in data)
            {
                result.Append(chars[b % (chars.Length)]);
            }
            return result.ToString();
        }
        private void frmProduct_Load(object sender, EventArgs e)
        {
            btnSave.Enabled = false;

            FillCombo();
            Autocomplete();
        }
        public void FillCombo()
        {
            try
            {

                con = new SqlConnection(cs.DBConn);
                con.Open();
                string ct = "SELECT RTRIM(CategoryName) from T_Category order by
CategoryName";
                cmd = new SqlCommand(ct);
```

```
                }
                con.Close();


                //////////////////////////// Select  Supplier ////////////

                con = new SqlConnection(cs.DBConn);
                con.Open();
                string sp = "SELECT SupplierID,CompanyName FROM T_SUPPLIERS ORDER BY
CompanyName";
                cmd = new SqlCommand(sp);
                cmd.Connection = con;
                rdr = cmd.ExecuteReader();

                while (rdr.Read())
                {
                    txtSupplierID.Text = rdr[0].ToString().Trim();
                    cmbSupplier.Items.Add(rdr[1]);
                }

                con.Close();

            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        private void Reset()
        {
            txtProductName.Text = "";
            cmbSubCategory.Text = "";
            cmbCategory.Text = "";
            txtFeatures.Text = "";
            pictureBox1.Image = Properties.Resources._12;

            cmbSubCategory.Enabled = false;
            btnDelete.Enabled = false;
            btnUpdate.Enabled = false;
            btnSave.Enabled = true;
            txtProductName.Focus();
        }

        private void btnNew_Click(object sender, EventArgs e)
        {
            Reset();
        }

        private void btnSave_Click(object sender, EventArgs e)
        {
            if (txtProductName.Text == "")
            {
                MessageBox.Show("Please enter product name", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                txtProductName.Focus();
                return;
            }
            if (cmbCategory.Text == "")
            {
                MessageBox.Show("Please select category", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                cmbCategory.Focus();
                return;
            }
```

```
                return;
            }
            if (cmbSupplier.Text == "")
            {
                MessageBox.Show("Please select Supplier", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                cmbSupplier.Focus();
                return;
            }

            try
            {
                con = new SqlConnection(cs.DBConn);
                con.Open();
                string ct = "SELECT ProductName FROM T_PRODUCT where ProductName='" +
txtProductName.Text + "'";

                cmd = new SqlCommand(ct);
                cmd.Connection = con;
                rdr = cmd.ExecuteReader();

                if (rdr.Read())
                {
                    MessageBox.Show("Product Name Already Exists", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                    txtProductName.Text = "";
                    txtProductName.Focus();


                    if ((rdr != null))
                    {
                        rdr.Close();
                    }
                    return;
                }
                auto();
                con = new SqlConnection(cs.DBConn);
                con.Open();
                string cb = "INSERT INTO
T_PRODUCT(ProductID,ProductName,CategoryID,SubCategoryID,SupplierID,Features,ProductEntry
Date,Image,Status) VALUES ('" + txtProductID.Text + "','" + txtProductName.Text + "'," +
txtCategoryID.Text + "," + txtSubCategoryID.Text + ",'" + txtSupplierID.Text.Trim() +
"',@d1,'"+DateTime.Now.Date.ToString()+"',@d2,1)";
                cmd = new SqlCommand(cb);
                cmd.Connection = con;
                cmd.Parameters.AddWithValue("@d1", txtFeatures.Text);
                MemoryStream ms = new MemoryStream();
                Bitmap bmpImage = new Bitmap(pictureBox1.Image);
                bmpImage.Save(ms, System.Drawing.Imaging.ImageFormat.Jpeg);
                byte[] data = ms.GetBuffer();
                SqlParameter p = new SqlParameter("@d2", SqlDbType.Image);
                p.Value = data;
                cmd.Parameters.Add(p);
                cmd.ExecuteReader();
                con.Close();
                MessageBox.Show("Successfully saved", "Record", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                Autocomplete();
                btnSave.Enabled = false;
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
```

```
            if (MessageBox.Show("Do you really want to delete this record?",
"Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
System.Windows.Forms.DialogResult.Yes)
            {
                delete_records();
            }
        }
        private void delete_records()
        {

            try
            {

                int RowsAffected = 0;
                con = new SqlConnection(cs.DBConn);
                con.Open();
                string cq = "DELETE FROM T_PRODUCT WHERE productID='" + txtProductID.Text
+ "'";
                cmd = new SqlCommand(cq);
                cmd.Connection = con;
                RowsAffected = cmd.ExecuteNonQuery();
                if (RowsAffected > 0)
                {
                    MessageBox.Show("Successfully deleted", "Record",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                    Reset();
                    Autocomplete();
                }
                else
                {
                    MessageBox.Show("No Record found", "Sorry", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                    Reset();
                    Autocomplete();
                }
                if (con.State == ConnectionState.Open)
                {
                    con.Close();
                }


            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
        private void Autocomplete()
        {
            try
            {
                con = new SqlConnection(cs.DBConn);
                con.Open();
                SqlCommand cmd = new SqlCommand("SELECT distinct ProductName FROM
T_PRODUCT", con);
                DataSet ds = new DataSet();
                SqlDataAdapter da = new SqlDataAdapter(cmd);
                da.Fill(ds, "Product");
                AutoCompleteStringCollection col = new AutoCompleteStringCollection();
                int i = 0;
                for (i = 0; i <= ds.Tables[0].Rows.Count - 1; i++)
                {
                    col.Add(ds.Tables[0].Rows[i]["productname"].ToString());

                }
```

```
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        private void btnUpdate_Click(object sender, EventArgs e)
        {
            if (txtProductName.Text == "")
            {
                MessageBox.Show("Please enter product name", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                txtProductName.Focus();
                return;
            }
            if (cmbCategory.Text == "")
            {
                MessageBox.Show("Please select category", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                cmbCategory.Focus();
                return;
            }
            if (cmbSubCategory.Text == "")
            {
                MessageBox.Show("Please select sub category", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                cmbSubCategory.Focus();
                return;
            }
            if (cmbSupplier.Text == "")
            {
                MessageBox.Show("Please select Supplier", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                cmbSupplier.Focus();
                return;
            }

            try
            {

                con = new SqlConnection(cs.DBConn);
                con.Open();
                string cb = "UPDATE T_PRODUCT set ProductName='" + txtProductName.Text +
"',CategoryID=" + txtCategoryID.Text + ",SubCategoryID=" + txtSubCategoryID.Text +
",SupplierID='"+txtSupplierID.Text+"',Features=@d1,Image=@d2 Where ProductID='" +
txtProductID.Text + "'";
                cmd = new SqlCommand(cb);
                cmd.Connection = con;
                cmd.Parameters.AddWithValue("@d1", txtFeatures.Text);
                MemoryStream ms = new MemoryStream();
                Bitmap bmpImage = new Bitmap(pictureBox1.Image);
                bmpImage.Save(ms, System.Drawing.Imaging.ImageFormat.Jpeg);
                byte[] data = ms.GetBuffer();
                SqlParameter p = new SqlParameter("@d2", SqlDbType.Image);
                p.Value = data;
                cmd.Parameters.Add(p);
                cmd.ExecuteReader();
                con.Close();
                MessageBox.Show("Successfully updated", "Record", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                Autocomplete();
                btnUpdate.Enabled = false;
            }
            catch (Exception ex)
```

```
        private void btnGetData_Click(object sender, EventArgs e)
        {
            this.Hide();
            frmProductsRecord2 frm = new frmProductsRecord2();
            frm.Show();
            frm.lblUserType.Text = lblUserType.Text;
            frm.GetData();
        }

        private void button1_Click(object sender, EventArgs e)
        {
              try
            {
                var _with1 = openFileDialog1;

                _with1.Filter = ("Image Files |*.png; *.bmp; *.jpg;*.jpeg; *.gif;");
                _with1.FilterIndex = 4;
                //Reset the file name
                openFileDialog1.FileName = "";

                if (openFileDialog1.ShowDialog() == DialogResult.OK)
                {
                    pictureBox1.Image = Image.FromFile(openFileDialog1.FileName);
                }

            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        //private void txtPrice_KeyPress(object sender, KeyPressEventArgs e)
        //{
        //     // allows 0-9, backspace, and decimal
        //    if (((e.KeyChar < 48 || e.KeyChar > 57) && e.KeyChar != 8 && e.KeyChar !=
46))
        //    {
        //        e.Handled = true;
        //        return;
        //    }

        //}

        private void cmbCategory_SelectedIndexChanged(object sender, EventArgs e)
        {
            try
            {
             con = new SqlConnection(cs.DBConn);

                con.Open();
                cmd = con.CreateCommand();

                cmd.CommandText = "SELECT ID FROM T_CATEGORY WHERE CategoryName = '" +
cmbCategory.Text + "'";
                rdr = cmd.ExecuteReader();

                if (rdr.Read())
                {
                    txtCategoryID.Text = rdr.GetInt32(0).ToString().Trim();
                }
                if ((rdr != null))
                {
                    rdr.Close();
```

```
            cmbCategory.Text = cmbCategory.Text.Trim();
            cmbSubCategory.Items.Clear();
            cmbSubCategory.Text = "";
            cmbSubCategory.Enabled = true;
            cmbSubCategory.Focus();

                con = new SqlConnection(cs.DBConn);
                con.Open();
                string ct = "SELECT distinct RTRIM(T_SUBCATEGORY.SubCategoryName) from
T_CATEGORY,T_SUBCATEGORY where T_CATEGORY.ID=T_SUBCATEGORY.CategoryID and
T_CATEGORY.CategoryName= '" + cmbCategory.Text + "'";
                cmd = new SqlCommand(ct);
                cmd.Connection = con;
                rdr = cmd.ExecuteReader();

                while (rdr.Read())
                {
                   cmbSubCategory.Items.Add(rdr[0]);
                }
                con.Close();

            }

            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }


        }

        private void cmbSubCategory_SelectedIndexChanged(object sender, EventArgs e)
        {
            //cmbSubCategory.Items.Clear();
            //cmbSubCategory.Text = "";

            try
            {
                con = new SqlConnection(cs.DBConn);

                con.Open();
                cmd = con.CreateCommand();

                cmd.CommandText = "SELECT ID FROM T_SUBCATEGORY WHERE SubCategoryName =
'" + cmbSubCategory.Text + "'";
                rdr = cmd.ExecuteReader();

                if (rdr.Read())
                {
                    txtSubCategoryID.Text = rdr.GetInt32(0).ToString().Trim();
                }
                if ((rdr != null))
                {
                    rdr.Close();
                }
                if (con.State == ConnectionState.Open)
                {
                    con.Close();
                }
                //cmbCategory.Text = cmbCategory.Text.Trim();
                //cmbSubCategory.Items.Clear();
                //cmbSubCategory.Text = "";
                //cmbSubCategory.Enabled = true;
                cmbSubCategory.Focus();
```

```
                cmd = new SqlCommand(ct);
                cmd.Connection = con;
                rdr = cmd.ExecuteReader();

                //cmbSubCategory.Items.Clear();
                while (rdr.Read())
                {
                    cmbSubCategory.Items.Add(rdr[0]);
                }
                con.Close();

            }

            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }

        }

        //private void txtSalePrice_KeyPress(object sender, KeyPressEventArgs e)
        //{
        //    if (((e.KeyChar < 48 || e.KeyChar > 57) && e.KeyChar != 8 && e.KeyChar !=
46))
        //    {
        //        e.Handled = true;
        //        return;
        //    }
        //}

        private void cmbSupplier_SelectedIndexChanged(object sender, EventArgs e)
        {
            try
            {
                con = new SqlConnection(cs.DBConn);

                con.Open();
                cmd = con.CreateCommand();

                cmd.CommandText = "SELECT SupplierID,CompanyName FROM T_SUPPLIERS WHERE
CompanyName = '" + cmbSupplier.Text + "'";
                rdr = cmd.ExecuteReader();

                 while(rdr.Read()==true)
                {
                    txtSupplierID.Text = rdr[0].ToString().Trim();
                    cmbSupplier.Text = rdr[1].ToString().Trim();
                }
                if ((rdr != null))
                {
                    rdr.Close();
                }
                if (con.State == ConnectionState.Open)
                {
                    con.Close();
                }
            }

            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
```

```
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Data.SqlClient;
using Excel = Microsoft.Office.Interop.Excel;


namespace Sales_and_Inventory_System
{
    public partial class frmStockIN : Form
    {
        SqlDataReader rdr = null;
        SqlConnection con = null;
        SqlCommand cmd = null;
        ConnectionString cs = new ConnectionString();
        public frmStockIN()
        {
            InitializeComponent();
        }
        public void GetData()
        {
            try
            {
                con = new SqlConnection(cs.DBConn);
                con.Open();
                String sql = "SELECT
T_Product.ProductID,T_Product.ProductName,T_Product.Features,T_STOCK.BuyPrice,T_STOCK.Sal
ePrice,sum(T_Temp_Stock.Quantity),sum(T_STOCK.SalePrice*T_Temp_Stock.Quantity) FROM
T_Temp_Stock,T_Product,T_STOCK WHERE T_Stock.ProductID=T_Product.ProductID AND
T_Temp_Stock.ProductID=T_Product.ProductID AND T_STOCK.StockID=T_Temp_Stock.StockID GROUP
BY
T_Product.productID,T_Product.productname,T_STOCK.BuyPrice,T_STOCK.SalePrice,T_PRODUCT.Fe
atures,T_Temp_Stock.Quantity having(T_Temp_Stock.Quantity>0)  order by ProductName";
                cmd = new SqlCommand(sql, con);
                rdr = cmd.ExecuteReader(CommandBehavior.CloseConnection);
                dataGridView1.Rows.Clear();
                while (rdr.Read() == true)
                {
                    dataGridView1.Rows.Add(rdr[0], rdr[1], rdr[2], rdr[3], rdr[4],
rdr[5],rdr[6]);
                }
                foreach (DataGridViewRow r in this.dataGridView1.Rows)
                {
                    if (Convert.ToInt32(r.Cells[4].Value) < 10)
                    {
                        r.DefaultCellStyle.BackColor = Color.Red;
                    }
                }
                con.Close();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
        private void frmStockIN_Load(object sender, EventArgs e)
        {
            GetData();
        }

        private void txtProductName_TextChanged(object sender, EventArgs e)
```

```
                String sql = "SELECT
Product.ProductID,ProductName,Features,BuyCost,SalePrice,sum(Quantity),sum(SalePrice*Quan
tity) from T_Temp_Stock,T_Product WHERE T_Temp_Stock.ProductID=T_Product.ProductID and
ProductName like '" + txtProductName.Text + "%' group by
product.ProductID,productname,BuyCost.SalePrice,Features,Quantity having(quantity>0)
order by ProductName";
                cmd = new SqlCommand(sql, con);
                rdr = cmd.ExecuteReader(CommandBehavior.CloseConnection);
                dataGridView1.Rows.Clear();
                while (rdr.Read() == true)
                {
                    dataGridView1.Rows.Add(rdr[0], rdr[1], rdr[2], rdr[3], rdr[4],
rdr[5],rdr[6]);
                }
                foreach (DataGridViewRow r in this.dataGridView1.Rows)
                {
                    if (Convert.ToInt32(r.Cells[4].Value) < 10)
                    {
                        r.DefaultCellStyle.BackColor = Color.Red;
                    }
                }
                con.Close();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        private void Button3_Click(object sender, EventArgs e)
        {
            int rowsTotal = 0;
            int colsTotal = 0;
            int I = 0;
            int j = 0;
            int iC = 0;
            System.Windows.Forms.Cursor.Current =
System.Windows.Forms.Cursors.WaitCursor;
            Excel.Application xlApp = new Excel.Application();

            try
            {
                Excel.Workbook excelBook = xlApp.Workbooks.Add();
                Excel.Worksheet excelWorksheet =
(Excel.Worksheet)excelBook.Worksheets[1];
                xlApp.Visible = true;

                rowsTotal = dataGridView1.RowCount - 1;
                colsTotal = dataGridView1.Columns.Count - 1;
                var _with1 = excelWorksheet;
                _with1.Cells.Select();
                _with1.Cells.Delete();
                for (iC = 0; iC <= colsTotal; iC++)
                {
                    _with1.Cells[1, iC + 1].Value = dataGridView1.Columns[iC].HeaderText;
                }
                for (I = 0; I <= rowsTotal - 1; I++)
                {
                    for (j = 0; j <= colsTotal; j++)
                    {
                        _with1.Cells[I + 2, j + 1].value =
dataGridView1.Rows[I].Cells[j].Value;
                    }
                }
                with1.Rows["1:1"].Font.FontStyle = "Bold";
```

```
                _with1.Cells[1, 1].Select();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
            finally
            {
                //RELEASE ALLOACTED RESOURCES
                System.Windows.Forms.Cursor.Current =
System.Windows.Forms.Cursors.Default;
                xlApp = null;
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            try
            {
                Cursor = Cursors.WaitCursor;
                timer1.Enabled = true;
                rptStock rpt = new rptStock();
                //The report you created.
                cmd = new SqlCommand();
                SqlDataAdapter myDA = new SqlDataAdapter();
                POS_DBDataSet myDS = new POS_DBDataSet();
                //The DataSet you created.
                con = new SqlConnection(cs.DBConn);
                cmd.Connection = con;
                cmd.CommandText = "SELECT
T_Product.ProductID,ProductName,Features,BuyCost,SalePrice,Quantity,Image from
T_Temp_Stock,T_Product where T_Temp_Stock.ProductID=T_Product.ProductID and ProductName
like '" + txtProductName.Text + "%' and quantity>0 order by ProductName";
                cmd.CommandType = CommandType.Text;
                myDA.SelectCommand = cmd;
                myDA.Fill(myDS, "T_temp_Stock");
                myDA.Fill(myDS, "T_product");
                rpt.SetDataSource(myDS);
                frmStockReport frm = new frmStockReport();
                frm.crystalReportViewer1.ReportSource = rpt;
                frm.Visible = true;
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            Cursor = Cursors.Default;
            timer1.Enabled = false;
        }

        private void button4_Click(object sender, EventArgs e)
        {
            try
            {
                Cursor = Cursors.WaitCursor;
                timer1.Enabled = true;
                rptStock rpt = new rptStock();
                //The report you created.
                cmd = new SqlCommand();
                SqlDataAdapter myDA = new SqlDataAdapter();
```

```
                cmd.CommandText = "SELECT
T_Product.ProductID,ProductName,Features,BuyCost,SalePrice,Quantity,BuyCost,SalePrice,Ima
ge from T_Temp_Stock,T_Product where T_Temp_Stock.ProductID=T_Product.ProductID and
quantity > 0 order by ProductName";
                cmd.CommandType = CommandType.Text;
                myDA.SelectCommand = cmd;
                myDA.Fill(myDS, "T_temp_Stock");
                myDA.Fill(myDS, "T_product");
                rpt.SetDataSource(myDS);
                frmStockReport frm = new frmStockReport();
                frm.crystalReportViewer1.ReportSource = rpt;
                frm.Visible = true;
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
```

CHAPTER 06

Summary and Conclusion



This chapter represents the summary, conclusion and recommendations of the present investigation.

Summary

The inclusion of Sales and Inventory System into a company provides a comprehensive sales and inventory system that is capable of providing and easy access sales and inventory conditions to customers and partners.

There are problems that encountered. One of it is that, their records don't coincide in their manually recorded data's. Another problem is that they often forgot to record the products that have been sold. The solution to these problems is to make automated or computerized sales and inventory system. Making this system can minimize their time period and can help them process their inventory easily. It is a hassle free and an easy access system. The system can Add, and Delete data's. It also has a transaction process so that in every product that the costumer will buy, will automatically recorded to the inventory. The flow on how the system works is that the user must input his or her username and password in for him or her to access the inventory, transaction process and adding and deleting of data. The inventory consist of the list of the products, the quantity and its price. While the transaction process consist of inputting data such as the Item name, Item price and the quantity of the product, also the amount received from the customer. By clicking the submit button, it will then, calculate the change and stock you can now save the data. After saving, it will automatically appear on the inventory sales.

CONCLUSION

"Sales and Inventory System" software developed for a company has been designed to achieve maximum efficiency and reduce the time taken to handle the Payroll activity. It is designed to replace an existing manual record system thereby reducing time taken for calculations and for storing data. The system uses Vb.Net as front end and Microsoft SQL as a backend for the database.

The system is strong enough to withstand regressive daily operations under conditions where the database is maintained and cleared over a certain time of span. The implementation of the system in the organization will considerably reduce data entry, time and also provide readily calculated reports.

7. References

(3) http://www.studymode.com/essays/Computer-Based-Inventory-System-401903.html

(4) http://www.studymode.com/essays/Sale-Inventory-System-1117762.html

(4) http://www.studymode.com/essays/Sales-And-Inventory-System-784556.html

(5) http://www.studymode.com/course-notes/Inventory-System-1128684.html

(6) http://www.studymode.com/subjects/related-studies-in-pos-page2.html

(7) http://www.studymode.com/essays/Chapter-Ii-Related-Literature-And-Studies-767850.html

(8) http://smallbusiness.chron.com/benefits-using-computerized-inventory-system-2044.html

(9) http://smallbusiness.chron.com/advantages-disadvantages-justintime-inventory-21407.html

(10)http://smallbusiness.chron.com/sales-inventory-monitoring-systems-18660.html

(11) https://valueline.com.ph/component/content/article/17-sme-solutions/14-vcheckout-pos.html

(12)http://www.medeil.com/English/philippines/

(13)http://www.asapsystems.com/barcloud/inventory-management-software-online.php

8. Future Objectives

Foremost, I would like to express my sincere gratitude to my honourable Teacher Md. Naderuzzaman for the continuous support of my study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me ia lot.
In addition I have add some of my purpose that I am going to implement in future to convenient and for user friendly interface of my project .They are followed below in order.

1.Notifification alert for the shortage of stock in menu tools
2 .Notification alert by a mobile SMS to the employee in charge of stock for any kind of stock shortage.
3.Develop a chatbot to chat internally inside the store among employees.
4. Develop a web server application for a runtime environment with 24/7 service.
5. Develop auto generated biiling and delivery service for any outbound order.